

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号
特開2004-213650
(P2004-213650A)

(43) 公開日 平成16年7月29日(2004.7.29)

(51) Int.Cl.⁷
G06F 12/14

F I
G O 6 F 12/14 3 2 0 B

テーマコード (参考)
5 B 0 1 7

審査請求 未請求 請求項の数 12 O L (全 34 頁)

(21) 出願番号	特願2003-423530 (P2003-423530)	(71) 出願人	399035766 エヌ・ティ・ティ・コミュニケーションズ株式会社 東京都千代田区内幸町一丁目1番6号
(22) 出願日	平成15年12月19日 (2003.12.19)	(74) 代理人	100083806 弁理士 三好 秀和
(31) 優先権主張番号	特願2002-367608 (P2002-367608)	(74) 代理人	100068342 弁理士 三好 保男
(32) 優先日	平成14年12月19日 (2002.12.19)	(74) 代理人	100095500 弁理士 伊藤 正和
(33) 優先権主張国	日本国 (JP)	(74) 代理人	100101247 弁理士 高橋 俊一
		(74) 代理人	100098327 弁理士 高松 俊雄

最終頁に続く

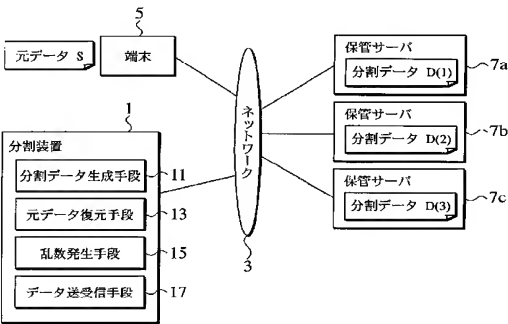
(54) 【発明の名称】 データ分割方法、データ分割装置およびコンピュータプログラム

(57) 【要約】

【課題】 比較的簡単な処理により元データを効率的に分割し得るデータ分割方法および装置を提供する。

【解決手段】 元データS、分割数n、処理単位ビット長bを設定し、元データSを処理単位ビット長b毎に区分けして複数の元部分データS(j)を生成し、複数の乱数部分データR(j)を生成し、各分割データD(i)を構成する各分割部分データ(i, j)を元部分データと乱数部分データの排他的論理和からなる所定の定義式に従って生成する。

【選択図】 図 1



【特許請求の範囲】

【請求項 1】

元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であって、

元データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、

この複数の元部分データの各々に対応して、元データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、

各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成し、

所望の分割数の分割データを複数の分割部分データから生成することにより、各分割データのみから元データが不明であるが、生成した分割データのうちの所定の個数の分割データから元データが復元可能であるようにする

ことを特徴とするデータ分割方法。

【請求項 2】

元部分データと乱数部分データは、分割数より 1 つ少ない複数個ずつ生成されることを特徴とする請求項 1 記載のデータ分割方法。

【請求項 3】

分割データは、乱数のみからなる 1 つ以上の分割データと、1 つ以上の元部分データと 1 つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる 1 つ以上の分割データを含むことを特徴とする請求項 1 記載のデータ分割方法。

【請求項 4】

乱数のみからなる 1 つの分割データは、任意に定めた長さの乱数を繰り返すことによって構成されることを特徴とする請求項 3 記載のデータ分割方法。

【請求項 5】

乱数のみからなる 1 つの分割データは、疑似乱数生成アルゴリズムに基づいて所定の長さの情報から生成された疑似乱数により構成されることを特徴とする請求項 3 または請求項 4 記載のデータ分割方法。

【請求項 6】

分割データは、1 つ以上の元部分データと 1 つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる 2 つ以上の分割データを含むことを特徴とする請求項 1 記載のデータ分割方法。

【請求項 7】

元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b を表すとともに、変数として $i(=1 \sim n)$ および $j(=1 \sim n-1)$ を用いて複数 $(n-1)$ 個の元部分データ、複数 $(n-1)$ 個の乱数部分データ、複数 (n) 個の分割データおよび各分割データの複数 $(n-1)$ 個の分割部分データのそれぞれのうちの 1 つをそれぞれ $S(j), R(j), D(i)$ および $D(i, j)$ で表わし、変数 j を 1 から $n-1$ まで変えて、各元部分データ $S(j)$ を元データ S の $b \times (j-1) + 1$ ビット目から b ビット分のデータとして作成し、 $U[n, n]$ を

$n \times n$ 行列で i 行 j 列の値 $u(i, j)$ が

$i+j \leq n+1$ のとき $u(i, j)=1$

$i+j > n+1$ のとき $u(i, j)=0$

である行列とし、 $P[n, n]$ を $n \times n$ 行列で i 行 j 列の値 $p(i, j)$ が

$j=i+1$ のとき $p(i, j)=1$

$i=1, j=n$ のとき $p(i, j)=1$

上記以外のとき $p(i, j)=0$

である行列としたとき、 $c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times P[n-1, n-1]^{(j-1)}$ の i 行 k 列の値と定義し、ただし $U[n-1, n-1] \times P[n-1, n-1]^{(j-1)}$ とは行列 $U[n-1, n-1]$ と $j-1$ 個の $P[n-1, n-1]$ の積を表し、 $Q(j, i, k)$ を $c(j, i, k)=1$ のとき、 $Q(j, i, k)=R(k)$ 、 $c(j, i, k)=0$ のとき、 $Q(j, i, k)=0$ と定義したとき、各分割部分データ $D(i, j)$ を、変数 i を 1 から n まで変えながら各変数 i において変数 j を 1 から $n-1$ まで変えて、 $i < n$ のとき、

40

50

【数 1】

$$D(i, j) = S(j) * \left(\prod_{k=1}^{n-1} Q(j, i, k) \right)$$

とし、 $i=n$ のとき、

$$D(i, j) = R(j)$$

として生成する、ただし

【数 2】

$$\prod_{k=1}^{n-1} Q(j, i, k) = Q(j, i, 1) * Q(j, i, 2) * \cdots * Q(j, i, n-1)$$

10

* は排他的論理和演算を表す、ことを特徴とする請求項 1 記載のデータ分割方法。

【請求項 8】

各分割データは、各分割データを構成する分割部分データ間で演算を行うことによって乱数成分が失われることがないように生成されることを特徴とする請求項 1 記載のデータ分割方法。

【請求項 9】

各分割データは、まず元部分データと乱数部分データの排他的論理和演算による所定の定義式を用いて各分割データを構成する複数の分割部分データを生成し、その後各分割データを構成するある 1 つの分割部分データと別の 1 つの分割部分データを入れ換えることによって生成されることを特徴とする請求項 8 記載のデータ分割方法。

20

【請求項 10】

各分割データは、まず元部分データと乱数部分データの排他的論理和演算による所定の定義式を用いて各分割データ $D(i)$ を構成する複数の分割部分データ $D(i, j)$ を生成し、その後各 i の値が $n-1 > i > 0$ である $D(i, j)$ から j 番目の乱数部分データ $R(j)$ を削除することによって生成される、ただし n は所望の分割数、 $j = (n-1) \times m + 1$ 、 $m \geq 0$ は任意の整数である、ことを特徴とする請求項 8 記載のデータ分割方法。

【請求項 11】

元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割装置であって、

30

元データを処理単位ビット長毎に区分けして、複数の元部分データを生成する元部分データ生成手段と、

この複数の元部分データの各々に対応して、元データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成する乱数生成手段と、

各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成する分割部分データ生成手段と、

所望の分割数の分割データを複数の分割部分データから生成することにより、各分割データのみから元データが不明であるが、生成した分割データのうちの所定の個数の分割データから元データが復元可能であるようにする分割データ生成手段と

40

を有することを特徴とするデータ分割装置。

【請求項 12】

元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割用のコンピュータプログラムであって、

元データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、

この複数の元部分データの各々に対応して、元データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、

各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成し、

所望の分割数の分割データを複数の分割部分データから生成することにより、各分割デ

50

ータのみから元データが不明であるが、生成した分割データのうちの所定の個数の分割データから元データが復元可能であるようにする

ことをコンピュータに実行させることを特徴としたコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データの機密性および安全性を確保するためにはデータを分割して保管することが有効であるが、このような場合などに有効なデータ分割方法および装置に関し、更に詳しくは、元データを所望の処理単位ビットに基づいて所望の分割数の分割データに分割するデータ分割方法および装置に関する。

10

【背景技術】

【0002】

重要な秘密データ（以下、元データという）を保管する場合、紛失、破壊、盗難やプライバシー侵害の脅威がある。このような脅威は秘密に保管すべきデータを単に暗号化しただけでは解決できず、紛失、破壊に備えてコピーを複数作ることが有効であるが、コピーを複数作ると盗難のリスクが増加してしまう。

【0003】

このような問題を解決する手段として、従来、しきい値秘密分散法がある（非特許文献1参照）。この従来の方法は、元データSをn個のデータに分割し、そのうち任意のx個の分割データを集めれば元データSが復元できるが、任意のx-1個の分割データでは元データSは復元できないというものである。従って、x-1個まで分割データが盗まれても元データSが漏れず、またn-x個まで分割データを紛失したり破壊されたりしても、元データSを復元できる。

20

【0004】

この方法の代表的な実現例としてn-1次多項式と剰余演算により構成される方法がある（非特許文献2参照）。この従来の方法は、公開鍵暗号方式の秘密鍵の分割管理などを利用してあり、データ量があまり多くないため、現状のコンピュータの演算処理能力、記憶装置・記憶媒体などのコストに対しては特に問題ない。

【非特許文献1】A. Shamir, "How to Share a Secret", Comm. Assoc. Comput. Mach., Vol. 22, no. 11, pp. 612-613 (Nov. 1979)

30

【非特許文献2】Bruce Schneier, "Applied Cryptography", John Wiley & Sons, Inc., p. 383-384 (1994)

【発明の開示】

【発明が解決しようとする課題】

【0005】

上述した従来の方法を安全に保管したいデータ量が例えばメガバイト、ギガバイトまたはそれ以上の規模となった場合に利用すると、多項式演算・剰余演算などを含む多倍長整数の演算処理を大量のデータに対して行う演算処理能力が必要となるとともに、またこの従来の方法では、例えば分割数n=5の場合には1バイトのデータから1バイトの分割データが5つ生成されるため、元データに対して単純に分割数に比例した倍数の記憶容量が必要となるなど、コンピュータを用いて具体的に実現する上で現実的ではないという問題がある。

40

【0006】

また、上述した従来の方法では、データの機密性を確保するため分割演算のための処理単位を設定しているが、この分割演算の処理単位にある程度のデータ長が必要となり、任意の処理単位で分割演算を行うことができないという問題もある。

【0007】

本発明は、上記に鑑みてなされたもので、その目的とするところは、比較的簡単な処理により元データを効率的に分割し得るデータ分割方法、装置およびコンピュータプログラムを提供することにある。

50

【課題を解決するための手段】

【0008】

上記目的を達成するため、本発明は、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であって、元データを処理単位ビット長毎に区別して、複数の元部分データを生成し、この複数の元部分データの各々に対応して、元データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成し、所望の分割数の分割データを複数の分割部分データから生成することにより、各分割データのみから元データが不明であるが、生成した分割データのうちの所定の個数の分割データから元データが復元可能であるようにすることを特徴とするデータ分割方法を提供する。

10

【0009】

また、本発明では、元部分データと乱数部分データは、分割数より1つ少ない複数個ずつ生成されることを特徴とする。

【0010】

また、本発明では、分割データは、乱数のみからなる1つ以上の分割データと、1つ以上の元部分データと1つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる1つ以上の分割データを含むことを特徴とする。

【0011】

また、本発明では、乱数のみからなる1つの分割データは、任意に定めた長さの乱数を繰り返すことによって構成されることを特徴とする。

20

【0012】

また、本発明では、乱数のみからなる1つの分割データは、疑似乱数生成アルゴリズムに基づいて所定の長さの情報から生成された疑似乱数により構成されることを特徴とする。

【0013】

また、本発明では、分割データは、1つ以上の元部分データと1つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる2つ以上の分割データを含むことを特徴とする。

【0014】

また、本発明は、元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b で表すとともに、変数として $i (=1 \sim n)$ および $j (=1 \sim n-1)$ を用いて複数 $(n-1)$ 個の元部分データ、複数 $(n-1)$ 個の乱数部分データ、複数 (n) 個の分割データおよび各分割データの複数 $(n-1)$ 個の分割部分データのそれぞれのうちの1つをそれぞれ $S(j), R(j), D(i)$ および $D(i, j)$ で表わし、変数 j を1から $n-1$ まで変えて、各元部分データ $S(j)$ を元データ S の $b \times (j-1) + 1$ ビット目から b ビット分のデータとして作成し、 $U[n, n]$ を $n \times n$ 行列で i 行 j 列の値 $u(i, j)$ が

$i+j \leq n+1$ のとき $u(i, j)=1$

$i+j > n+1$ のとき $u(i, j)=0$

である行列とし、 $P[n, n]$ を $n \times n$ 行列で i 行 j 列の値 $p(i, j)$ が

$j=i+1$ のとき $p(i, j)=1$

$i=1, j=n$ のとき $p(i, j)=1$

上記以外のとき $p(i, j)=0$

である行列としたとき、 $c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times P[n-1, n-1]^{(j-1)}$ の i 行 k 列の値と定義し、ただし $U[n-1, n-1] \times P[n-1, n-1]^{(j-1)}$ とは行列 $U[n-1, n-1]$ と $j-1$ 個の $P[n-1, n-1]$ の積を表し、 $Q(j, i, k)$ を $c(j, i, k)=1$ のとき、 $Q(j, i, k)=R(k)$ 、 $c(j, i, k)=0$ のとき、 $Q(j, i, k)=0$ と定義したとき、各分割部分データ $D(i, j)$ を、変数 i を1から n まで変えながら各変数 i において変数 j を1から $n-1$ まで変えて、 $i < n$ のとき、

30

40

【数 3】

$$D(i, j) = S(j) * \left(\prod_{k=1}^{n-1} Q(j, i, k) \right)$$

【0015】

とし、 $i=n$ のとき、

$$D(i, j) = R(j)$$

として生成する、ただし

【数 4】

$$\prod_{k=1}^{n-1} Q(j, i, k) = Q(j, i, 1) * Q(j, i, 2) * \cdots * Q(j, i, n-1)$$

10

【0016】

* は排他的論理和演算を表す、ことを特徴とする。

【0017】

また、本発明では、各分割データは、各分割データを構成する分割部分データ間で演算を行うことによって乱数成分が失われることがないように生成されることを特徴とする。

【0018】

また、本発明では、各分割データは、まず元部分データと乱数部分データの排他的論理和演算による所定の定義式を用いて各分割データを構成する複数の分割部分データを生成し、その後各分割データを構成するある1つの分割部分データと別の1つの分割部分データを入れ換えることによって生成されることを特徴とする。

20

【0019】

また、本発明では、各分割データは、まず元部分データと乱数部分データの排他的論理和演算による所定の定義式を用いて各分割データ $D(i)$ を構成する複数の分割部分データ $D(i, j)$ を生成し、その後各 i の値が $n-1 > i > 0$ である $D(i, j)$ から j 番目の乱数部分データ $R(j)$ を削除することによって生成される、ただし n は所望の分割数、 $j = (n-1) \times m + 1$ 、 $m \geq 0$ は任意の整数である、ことを特徴とする。

【0020】

さらに、本発明は、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割装置であって、元データを処理単位ビット長毎に区分けして、複数の元部分データを生成する元部分データ生成手段と、この複数の元部分データの各々に対応して、元データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成する乱数生成手段と、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成する分割部分データ生成手段と、所望の分割数の分割データを複数の分割部分データから生成することにより、各分割データのみから元データが不明であるが、生成した分割データのうちの所定の個数の分割データから元データが復元可能であるようにする分割データ生成手段とを有することを特徴とするデータ分割装置を提供する。

30

【0021】

さらに、本発明は、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割用のコンピュータプログラムであって、元データを処理単位ビット長毎に区分けして、複数の元部分データを生成し、この複数の元部分データの各々に対応して、元データのビット長と同じまたはこれより短い長さの乱数から処理単位ビット長の複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和によって処理単位ビット長毎に生成し、所望の分割数の分割データを複数の分割部分データから生成することにより、各分割データのみから元データが不明であるが、生成した分割データのうちの所定の個数の分割データから元データが復元可能であるようにすることをコンピュータに実行させることを特徴としたコンピュータプログラムを提供する。

40

50

【発明の効果】

【0022】

以上説明したように、本発明によれば、元データを処理単位ビット長毎に区分けして複数の元部分データを生成し、複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和からなる所定の定義式に従って生成するので、従来のように多項式や剰余演算を用いることなく、コンピュータ処理に適したビット演算である排他的論理和演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができるとともに、また分割データの保管に必要な記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。

10

【発明を実施するための最良の形態】

【0023】

以下、図面を用いて本発明の実施の形態を説明する。図1は、本発明の一実施形態に係るデータ分割方法を実施するデータ分割装置を含むシステム構成図である。本実施形態のデータ分割装置は、符号1で示すように分割装置1としてネットワーク3に接続されて設けられ、このネットワーク3にアクセスしてくる端末5からの元データ分割要求に応じて元データ8を複数の分割データに分割し、この分割した複数の分割データをネットワーク3を介して複数の保管サーバ7a、7b、7cに保管するようになっている。なお、図1では、分割装置1は、端末5からの元データ8を3つの分割データD(1)、D(2)、D(3)に分割し、それぞれを複数の保管サーバ7a、7b、7cに保管するようになっている。

20

【0024】

また、分割装置1は、ネットワーク3を介してアクセスしてくる端末5からの元データ復元要求に応じて複数の分割データD(1)、D(2)、D(3)をネットワーク3を介して各保管サーバ7から取得し、この取得した複数の分割データD(1)、D(2)、D(3)から元データ8を復元し、ネットワーク3を介して端末5に送信するようになっている。

【0025】

分割装置1は、詳しくは、元データ8から複数の分割データDを生成する分割データ生成手段11、複数の分割データDから元データ8を復元する元データ復元手段13、元データ8から複数の分割データDを生成するために使用される乱数Rを発生する乱数発生手段15、および分割データ生成手段11で生成した複数の分割データDをネットワーク3を介して複数の保管サーバ7a、7b、7cに送信したり、また複数の保管サーバ7a、7b、7cからの複数の分割データDをネットワーク3を介して受信するためのデータ送受信手段17から構成されている。

30

【0026】

上述したように構成される本実施形態における元データの分割および復元では、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するが、この場合の処理単位ビット長は任意の値に設定することができ、元データを処理単位ビット長毎に区分けして、この元部分データから分割部分データを分割数より1少ない数ずつ生成するので、元データのビット長が処理単位ビット長の(分割数1)倍の整数倍に一致しない場合は、元データの末尾の部分に0を埋めるなどして元データのビット長を処理単位ビット長の(分割数1)倍の整数倍に合わせることで本実施形態を適用することができる。

40

【0027】

また、上述した乱数も(分割数1)個の元部分データの各々に対応して処理単位ビット長のビット長を有する(分割数1)個の乱数部分データとして乱数発生手段15から生成される。すなわち、乱数は処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する(分割数1)個の乱数部分データとして生成される。更に、元データは処理単位ビット長に基づいて所望の分割数の分割データに分割されるが、この分割データの各々も(分割数1)個の元部分データの各々に対応して処理単位ビット長のビット長を有する(分割数1)個の分割部分データとして生成される。すなわち、分割データの各々は、

50

処理単位ビット長毎に区分けられて、処理単位ビット長のビット長を有する（分割数 1）個の分割部分データとして生成される。

【0028】

なお、以下の説明では、上述した元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b で表すとともに、また複数のデータや乱数などのうちの 1 つを表わす変数として $i (=1 \sim n)$ および $j (=1 \sim n-1)$ を用い、（分割数 $n-1$ ）個の元部分データ、（分割数 $n-1$ ）個の乱数部分データ、および分割数 n 個の分割データ D のそれぞれのうちの 1 つをそれぞれ $S(j), R(j)$ および $D(i)$ で表記し、更に各分割データ $D(i)$ を構成する複数 $(n-1)$ の分割部分データを $D(i, j)$ で表記するものとする。すなわち、 $S(j)$ は、元データ S の先頭から処理単位ビット長毎に区分けして 1 番から順に採番した時の j 番目の元部分データを表すものである。

10

【0029】

この表記を用いると、元データ、乱数データ、分割データとこれらをそれぞれ構成する元部分データ、乱数部分データ、分割部分データは、次のように表記される。

【0030】

元データ $S = (n-1)$ 個の元部分データ $S(j)$
 $= S(1), S(2), \dots, S(n-1)$
 乱数 $R = (n-1)$ 個の乱数部分データ $R(j)$
 $= R(1), R(2), \dots, R(n-1)$
 n 個の分割データ $D(i) = D(1), D(2), \dots, D(n)$
 各分割部分データ $D(i, j)$
 $= D(1, 1), D(1, 2), \dots, D(1, n-1)$
 $D(2, 1), D(2, 2), \dots, D(2, n-1)$
 \dots
 $D(n, 1), D(n, 2), \dots, D(n, n-1)$
 $(i=1 \sim n), (j=1 \sim n-1)$

20

本実施形態は、上述したように処理単位ビット長毎に区分けされる複数の部分データに対して元部分データと乱数部分データの排他的論理和演算 (XOR) を行って、詳しくは、元部分データと乱数部分データの排他的論理和演算 (XOR) からなる定義式を用いて、元データの分割を行うことを特徴とするものであり、上述したデータ分割処理に多項式や剰余演算を用いる従来の方法に比較して、コンピュータ処理に適したビット演算である排他的論理和 (XOR) 演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを生成することができるとともに、また分割データの保管に必要な記憶容量も分割数に比例した倍数の容量よりも小さくすることができ。更に、任意に定められた一定の長さ毎にデータの先頭から順に演算処理を行うストリーム処理により分割データが生成される。

30

【0031】

なお、本実施形態で使用する排他的論理和演算 (XOR) は、以下の説明では、「*」なる演算記号で表すことにするが、この排他的論理和演算のビット毎の演算規則での各演算結果は下記のとおりである。

40

【0032】

0 * 0 の演算結果は 0
 0 * 1 の演算結果は 1
 1 * 0 の演算結果は 1
 1 * 1 の演算結果は 0

また、XOR 演算は交換法則、結合法則が成り立つ。すなわち、

$a * b = b * a$
 $(a * b) * c = a * (b * c)$

が成り立つことが数学的に証明される。

【0033】

50

また、 $a*a=0$ 、 $a*0=0*a=a$ が成り立つ。

【0034】

ここで a, b, c は同じ長さのビット列を表し、0はこれらと同じ長さですべて「0」からなるビット列を表す。

【0035】

次に、フローチャートなどの図面も参照して、上記実施形態の作用について説明するが、この説明の前に図2、図5、図8、図9のフローチャートに示す記号の定義について説明する。

【0036】

(1)

10

【数5】

$$(1) \prod_{i=1}^n A(i)$$

【0037】

は、 $A(1)*A(2)*\dots*A(n)$ を意味するものとする。

【0038】

(2) $c(j, i, k)$ を、 $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times (P[n-1, n-1])^{(j-1)}$ の i 行 k 列の値と定義する。

【0039】

20

このとき $Q(j, i, k)$ を下記のように定義する。

【0040】

$$c(j, i, k)=1 \text{ のとき } Q(j, i, k)=R((n-1) \times m+k)$$

$$c(j, i, k)=0 \text{ のとき } Q(j, i, k)=0$$

ただし、 m は $m \geq 0$ の整数を表す。

【0041】

(3) $U[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $u(i, j)$ を表すと、

$$i+j \leq n+1 \text{ のとき } u(i, j)=1$$

$$i+j > n+1 \text{ のとき } u(i, j)=0$$

である行列を意味するものとし、「上三角行列」ということとする。具体的には下記のような行列である。

30

【数6】

$$U[3, 3] = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

$$U[4, 4] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

40

【0042】

(4) $P[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $p(i, j)$ を表すと、

$$j=i+1 \text{ のとき } p(i, j)=1$$

$$i=1, j=n \text{ のとき } p(i, j)=1$$

$$\text{上記以外のとき } p(i, j)=0$$

である行列を意味するものとし、「回転行列」ということとする。具体的には下記のような

50

な行列であり、他の行列の右側からかけると当該他の行列の1列目を2列目へ、2列目を3列目へ、...、n-1列目をn列目へ、n列目を1列目へ移動させる作用がある。つまり、行列Pを他の行列に右側から複数回かけると、その回数分だけ各列を右方向へ回転させるように移動させることができる。

【数7】

$$P[3,3] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

10

$$P[4,4] = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

【0043】

(5) A, Bをn×n行列とすると、A×Bとは行列AとBの積を意味するものとする。行列の成分同士の計算規則は通常の数学で用いるものと同じである。

【0044】

(6) Aをn×n行列とし、iを整数とすると、Aⁱとは行列Aのi個の積を意味するものとする。また、A⁰とは単位行列Eを意味するものとする。

20

【0045】

(7) 単位行列E[n, n]とは、n×n行列であって、i行j列の値をe(i, j)を表すと、

i=j のとき e(i, j)=1

上記以外のとき e(i, j)=0

である行列を意味するものとする。具体的には下記のような行列である。Aを任意のn×n行列とすると

$$A \times E = E \times A = A$$

となる性質がある。

30

【数8】

$$E[3,3] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E[4,4] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

40

【0046】

次に、図2に示すフローチャートおよび図3、図4に示す具体的データなどを参照して、上記実施形態の作用として、まず元データ8の分割処理について説明する。

【0047】

本実施形態の分割装置1の利用者は、端末5からネットワーク3を介して分割装置1にアクセスし、分割装置1に元データ8を送信し、分割装置1ではデータ送受信手段17が端末5からの元データ8を受信し、分割装置1に供給する(図2のステップ8201)。なお、本例では、元データ8は、16ビットの「10110010 00110111」とする。

50

【0048】

次に、利用者は端末5から分割数 n として8を分割装置1に指示する（ステップS203）。この分割数 n は分割装置1において予め定められた値を用いてもよい。なお、この分割数 $n=8$ に従って分割装置1で生成される8個の分割データを $D(1)$ 、 $D(2)$ 、 $D(3)$ とする。この分割データ $D(1)$ 、 $D(2)$ 、 $D(3)$ は、すべて元データのビット長と同じ16ビット長のデータである。

【0049】

それから、元データ8を分割するために使用される処理単位ビット長 b を8ビットと決定し、また元データと同じビット長である16ビットの乱数 R を乱数発生手段15から取得して生成する（ステップS205）。この処理単位ビット長 b は、利用者が端末5から分割装置1に対して指定してもよいし、または分割装置1において予め定められた値を用いてもよい。なお、処理単位ビット長 b は、任意のビット数でよいが、ここでは元データ8を割り切れることができる8ビットとしている。従って、上記16ビットの「10110010 00110111」の元データ8は、8ビットの処理単位ビット長で分けられた場合の2個の元分割データ $S(1)$ および $S(2)$ は、それぞれ「10110010」および「00110111」となる。

【0050】

次のステップS207では、元データ8のビット長が 8×2 の整数倍であるか否かを判定し、整数倍でない場合には、元データ8の末尾を0で埋めて、 8×2 の整数倍に合わせる。なお、本例のように処理単位ビット長 b が8ビットおよび分割数 n が8に設定された場合における分割処理は、元データ8のビット長として16ビットに限られるものでなく、処理単位ビット長 $b \times (\text{分割数} n - 1) = 8 \times 2$ の整数倍の元データ8に対して有効なものである。

【0051】

次に、ステップS209では、変数 m 、すなわち上述した整数倍を意味する変数 m を0に設定する。本例のように、元データ8が処理単位ビット長 $b \times (\text{分割数} n - 1) = 8 \times 2 = 16$ ビットである場合には、変数 m は0であるが、2倍の32ビットの場合には、変数 m は1となり、3倍の48ビットの場合には、変数 m は2となる。

【0052】

次に、元データ8の $8 \times 2 \times m + 1$ ビット目から 8×2 ビット分のデータが存在するか否かが判定される（ステップS211）。これは、このステップS211以降に示す分割処理を元データ8の変数 m で特定される処理単位ビット長 $b \times (\text{分割数} n - 1) = 8 \times 2 = 16$ ビットに対して行った後、元データ8として次の16ビットがあるか否かを判定しているものである。本例のように元データ8が16ビットである場合には、16ビットの元データ8に対してステップS211以降の分割処理を1回行うと、後述するステップS219で変数 m が+1されるが、本例の元データ8では変数 m が $m+1$ の場合に相当する17ビット以降のデータは存在しないので、ステップS211からステップS221に進むことになるが、今の場合は、変数 m は0であるので、元データ8の $8 \times 2 \times m + 1$ ビット目は、 $8 \times 2 \times 0 + 1 = 1$ となり、元データ8の16ビットの1ビット目から 8×2 ビット分にデータが存在するため、ステップS213に進む。

【0053】

ステップS213では、変数 j を1から2（=分割数 $n - 1$ ）まで変えて、元データ8の $8 \times (2 \times m + j - 1) + 1$ ビット目から8ビット分（=処理単位ビット長）のデータを元部分データ $S(2 \times m + j)$ に設定し、これにより元データ8を処理単位ビット長で分けした2（分割数 $n - 1$ ）個の元部分データ $S(1)$ 、 $S(2)$ を次のように生成する。

【0054】

元データ8= $S(1)$ 、 $S(2)$

第1の元部分データ $S(1)$ = 「10110010」

第2の元部分データ $S(2)$ = 「00110111」

次に、変数 j を1から2（=分割数 $n - 1$ ）まで変えて、乱数部分データ $R(2 \times m + j)$ に乱数発生手段15から発生する8ビットの長さの乱数を設定し、これにより乱数 R を処理単位ビット長で分けした2（分割数 $n - 1$ ）個の乱数部分データ $R(1)$ 、 $R(2)$ を次のように生成する（ス

10

20

30

40

50

テ ッ プ S 2 1 5) 。

【 0 0 5 5 】

乱数 $R=R(1), R(2)$

第 1 の乱数部分データ $R(1)=「10110001」$

第 2 の乱数部分データ $R(2)=「00110101」$

次に、ステップ S 2 1 7 において、変数 i を 1 から 3 (=分割数 n) まで変えるとともに、更に各変数 i において変数 j を 1 から 2 (=分割数 $n-1$) まで変えながら、ステップ S 2 1 7 に示す分割データを生成するための元部分データと乱数部分データの排他的論理和からなる定義式により複数の分割データ $D(i)$ の各々を構成する各分割部分データ $D(i, 2 \times m + j)$ を生成する。この結果、次に示すような分割データ D が生成される。

10

【 0 0 5 6 】

分割データ D

= 3 個の分割データ $D(i)=D(1), D(2), D(3)$

第 1 の分割データ $D(1)$

= 2 個の分割部分データ $D(1, j)=D(1, 1), D(1, 2)$

= 「00110110」, 「10110011」

第 2 の分割データ $D(2)$

= 2 個の分割部分データ $D(2, j)=D(2, 1), D(2, 2)$

= 「00000011」, 「00000010」

第 3 の分割データ $D(3)$

= 2 個の分割部分データ $D(3, j)=D(3, 1), D(3, 2)$

= 「10110001」, 「00110101」

20

なお、各分割部分データ $D(i, j)$ を生成するためのステップ S 2 1 7 に示す定義式は、本例のように分割数 $n=3$ の場合には、具体的には図 4 に示す表に記載されているものとなる。図 4 に示す表から、分割部分データ $D(1, 1)$ を生成するための定義式は $S(1)*R(1)*R(2)$ であり、 $D(1, 2)$ の定義式は $S(2)*R(1)*R(2)$ であり、 $D(2, 1)$ の定義式は $S(1)*R(1)$ であり、 $D(2, 2)$ の定義式は $S(2)*R(2)$ であり、 $D(3, 1)$ の定義式は $R(1)$ であり、 $D(3, 2)$ の定義式は $R(2)$ である。また、図 4 に示す表には $m>0$ の場合の任意の整数についての一般的な定義式も記載されている。

【 0 0 5 7 】

このように整数倍を意味する変数 $m=0$ の場合について分割データ D を生成した後、次に変数 m を 1 増やし (ステップ S 2 1 9)、ステップ S 2 1 1 に戻り、変数 $m+1$ に該当する元データ S の 17 ビット以降について同様の分割処理を行おうとするが、本例の元データ S は 16 ビットであり、17 ビット以降のデータは存在しないので、ステップ S 2 1 1 からステップ S 2 2 1 に進み、上述したように生成した分割データ $D(1), D(2), D(3)$ を分割装置 1 のデータ送受信手段 17 からネットワーク 3 を介して保管サーバ 7a, 7b, 7c にそれぞれ送信し、各保管サーバ 7 に保管し、分割処理を終了する。

30

【 0 0 5 8 】

ここで、上述した図 2 のフローチャートのステップ S 2 1 7 における定義式による分割データの生成処理、具体的には分割数 $n=3$ の場合の分割データの生成処理について詳しく説明する。

40

【 0 0 5 9 】

まず、整数倍を意味する変数 $m=0$ の場合には、ステップ S 2 1 7 に示す定義式から各分割データ $D(i)=D(1) \sim D(3)$ の各々を構成する各分割部分データ $D(i, 2 \times m + j)=D(i, j) (i=1 \sim 3, j=1 \sim 2)$ は、次のようになる。

【 0 0 6 0 】

$D(1, 1)=S(1)*Q(1, 1, 1)*Q(1, 1, 2)$

$D(1, 2)=S(2)*Q(2, 1, 1)*Q(2, 1, 2)$

$D(2, 1)=S(1)*Q(1, 2, 1)*Q(1, 2, 2)$

$D(2, 2)=S(2)*Q(2, 2, 1)*Q(2, 2, 2)$

50

$$D(3, 1)=R(1)$$

$$D(3, 2)=R(2)$$

上記の6つの式のうち上から4つの式に含まれる $Q(j, i, k)$ を具体的に求める。

【0061】

これは $c(j, i, k)$ を 2×2 行列である $U[2, 2] \times (P[2, 2])^{(j-1)}$ の i 行 k 列の値としたとき下記のように定義される。

【0062】

$$c(j, i, k)=1 \text{ のとき } Q(j, i, k)=R(k)$$

$$c(j, i, k)=0 \text{ のとき } Q(j, i, k)=0$$

ここで、

$j=1$ のときは

【数9】

$$\begin{aligned} U[2, 2] \times (P[2, 2])^{(j-1)} &= U[2, 2] \times (P[2, 2])^0 \\ &= U[2, 2] \times E[2, 2] \\ &= U[2, 2] \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \end{aligned}$$

10

【0063】

$j=2$ のときは

【数10】

$$\begin{aligned} U[2, 2] \times (P[2, 2])^{(j-1)} &= U[2, 2] \times (P[2, 2])^1 \\ &= U[2, 2] \times P[2, 2] \\ &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

20

30

【0064】

これを用いると、各分割部分データ $D(i, j)$ は次のような定義式により生成される。

【0065】

$$D(1, 1)=S(1)*Q(1, 1, 1)*Q(1, 1, 2)=S(1)*R(1)*R(2)$$

$$D(1, 2)=S(2)*Q(2, 1, 1)*Q(2, 1, 2)=S(2)*R(1)*R(2)$$

$$D(2, 1)=S(1)*Q(1, 2, 1)*Q(1, 2, 2)=S(1)*R(1)*0=S(1)*R(1)$$

$$D(2, 2)=S(2)*Q(2, 2, 1)*Q(2, 2, 2)=S(2)*0*R(2)=S(2)*R(2)$$

上述した各分割部分データ $D(i, j)$ を生成するための定義式は、図3にも図示されている。

40

【0066】

図3は、上述したように16ビットの元データ S を8ビットの処理単位ビット長に基づいて分割数 $n=3$ で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【0067】

ここで、上述した定義式により分割データ $D(1)$ 、 $D(2)$ 、 $D(3)$ および各分割部分データ $D(1, 1)$ 、 $D(1, 2)$ 、 $D(2, 1)$ 、 $D(2, 2)$ 、 $D(3, 1)$ 、 $D(3, 2)$ を生成する過程と定義式の一般形について説明する。

【0068】

50

まず、第1の分割データD(1)に対しては、第1の分割部分データD(1,1)は、上述した定義式 $S(1)*R(1)*R(2)$ で定義され、第2の分割部分データD(1,2)は定義式 $S(2)*R(1)*R(2)$ で定義される。なお、この定義式の一般形は、D(1,j)に対しては $S(j)*R(j)*R(j+1)$ であり、D(1,j+1)に対して $S(j+1)*R(j)*R(j+1)$ である(jは奇数とする)。定義式に従って計算すると、D(1,1)は00110110、D(1,2)は10110011となるので、D(1)は00110110 10110011である。なお、定義式の一般形は、図4にまとめて示されている。

【0069】

また、第2の分割データD(2)に対しては、D(2,1)は $S(1)*R(1)$ で定義され、D(2,2)は $S(2)*R(2)$ で定義される。この定義式の一般形は、D(2,j)に対しては $S(j)*R(j)$ であり、D(2,j+1)に対しては $S(j+1)*R(j+1)$ である(jは奇数とする)。定義式に従って計算すると、D(2,1)は00000011、D(2,2)は00000010となるので、D(2)は00000011 00000010である。

【0070】

更に第3の分割データD(3)に対しては、D(3,1)は $R(1)$ で定義され、D(3,2)は $R(2)$ で定義される。この定義式の一般形は、D(3,j)に対しては $R(j)$ であり、D(3,j+1)に対しては $R(j+1)$ である(jは奇数とする)。定義式に従って計算すると、D(3,1)は10110001、D(3,2)は00110101となるので、D(3)は10110001 00110101である。

【0071】

上記説明は、S、R、D(1)、D(2)、D(3)の長さを16ビットとしたが、データの先頭から上記分割処理を繰り返すことにより、どのような長さの元データSからでも分割データD(1)、D(2)、D(3)を生成することができる。また、処理単位ビット長bは任意にとることができ、元データSの先頭から順に $b \times 2$ の長さ毎に上記分割処理を繰り返すことにより任意の長さの元データ、具体的には処理単位ビット長 $b \times 2$ の整数倍の長さの元データに対して適用することができる。なお、元データSの長さが処理単位ビット長 $b \times 2$ の整数倍でない場合は、例えば、データ末尾の部分を0で埋めるなどして元データSの長さを処理単位ビット長 $b \times 2$ の整数倍に合わせることで上記した本実施形態の分割処理を適用することができる。

【0072】

次に、図3の右側に示す表を参照して、分割データから元データを復元する処理について説明する。

【0073】

まず、利用者は端末5からネットワーク3を介して分割装置1にアクセスし、分割装置1のデータ送受信手段17を介して元データSの復元を要求する。分割装置1は、この元データSの復元要求を受け取ると、この元データSに対応する分割データD(1)、D(2)、D(3)が保管サーバ7a、7b、7cに保管されていることを記憶しているため、ネットワーク3を介して保管サーバ7a、7b、7cから分割データD(1)、D(2)、D(3)を取得し、この取得した分割データD(1)、D(2)、D(3)から次に示すように元データSを復元する。

【0074】

まず、分割部分データD(2,1)、D(3,1)から第1の元部分データS(1)を次のように生成することができる。

【0075】

$$\begin{aligned} D(2,1)*D(3,1) &= (S(1)*R(1))*R(1) \\ &= S(1)*(R(1)*R(1)) \\ &= S(1)*0 \\ &= S(1) \end{aligned}$$

具体的に計算すると、D(2,1)は00000011、D(3,1)は10110001なので、S(1)は10110010となる。

【0076】

また、別の分割部分データから次のように第2の元部分データS(2)を生成することができる。

【0077】

10

20

30

40

50

$$\begin{aligned}
 D(2, 2) * D(3, 2) &= (S(2) * R(2)) * R(2) \\
 &= S(2) * (R(2) * R(2)) \\
 &= S(2) * 0 \\
 &= S(2)
 \end{aligned}$$

具体的に計算すると、 $D(2, 2)$ は00000010、 $D(3, 2)$ は00110101なので、 $S(2)$ は00110111となる。

【0078】

一般に、 j を奇数として、

$$\begin{aligned}
 D(2, j) * D(3, j) &= (S(j) * R(j)) * R(j) \\
 &= S(j) * (R(j) * R(j)) \\
 &= S(j) * 0 \\
 &= S(j)
 \end{aligned}$$

10

であるから、 $D(2, j) * D(3, j)$ を計算すれば、 $S(j)$ が求まる。

【0079】

また、一般に、 j を奇数として、

$$\begin{aligned}
 D(2, j+1) * D(3, j+1) &= (S(j+1) * R(j+1)) * R(j+1) \\
 &= S(j+1) * (R(j+1) * R(j+1)) \\
 &= S(j+1) * 0 \\
 &= S(j+1)
 \end{aligned}$$

であるから、 $D(2, j+1) * D(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

20

【0080】

次に、 $D(1)$ 、 $D(3)$ を取得して S を復元する場合には、次のようになる。

【0081】

$$\begin{aligned}
 D(1, 1) * D(3, 1) * D(3, 2) &= (S(1) * R(1) * R(2)) * R(1) * R(2) \\
 &= S(1) * (R(1) * R(1)) * (R(2) * R(2)) \\
 &= S(1) * 0 * 0 \\
 &= S(1)
 \end{aligned}$$

であるから、 $D(1, 1) * D(3, 1) * D(3, 2)$ を計算すれば、 $S(1)$ が求まる。具体的に計算すると、 $D(1, 1)$ は00110110、 $D(3, 1)$ は10110001、 $D(3, 2)$ は00110101なので、 $S(1)$ は10110010となる。

30

【0082】

また同様に、

$$\begin{aligned}
 D(1, 2) * D(3, 1) * D(3, 2) &= (S(2) * R(1) * R(2)) * R(1) * R(2) \\
 &= S(2) * (R(1) * R(1)) * (R(2) * R(2)) \\
 &= S(2) * 0 * 0 \\
 &= S(2)
 \end{aligned}$$

であるから、 $D(1, 2) * D(3, 1) * D(3, 2)$ を計算すれば、 $S(2)$ が求まる。具体的に計算すると、 $D(1, 2)$ は10110011、 $D(3, 1)$ は10110001、 $D(3, 2)$ は00110101なので、 $S(2)$ は00110111となる。

【0083】

一般に、 j を奇数として、

$$\begin{aligned}
 D(1, j) * D(3, j) * D(3, j+1) &= (S(j) * R(j) * R(j+1)) * R(j) * R(j+1) \\
 &= S(j) * (R(j) * R(j)) * (R(j+1) * R(j+1)) \\
 &= S(j) * 0 * 0 \\
 &= S(j)
 \end{aligned}$$

であるから、 $D(1, j) * D(3, j) * D(3, j+1)$ を計算すれば、 $S(j)$ が求まる。

40

【0084】

また、一般に、 j を奇数として、

$$\begin{aligned}
 D(1, j+1) * D(3, j) * D(3, j+1) &= (S(j+1) * R(j) * R(j+1)) * R(j) * R(j+1) \\
 &= S(j+1) * (R(j) * R(j)) * (R(j+1) * R(j+1))
 \end{aligned}$$

50

$$\begin{aligned}
 &= S(j+1) * 0 * 0 \\
 &= S(j+1)
 \end{aligned}$$

であるから、 $D(1, j+1) * D(3, j) * D(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0085】

次に、 $D(1), D(2)$ を取得して S を復元する場合には、次のようになる。

【0086】

$$\begin{aligned}
 D(1, 1) * D(2, 1) &= (S(1) * R(1) * R(2)) * (S(1) * R(1)) \\
 &= (S(1) * S(1)) * (R(1) * R(1)) * R(2) \\
 &= 0 * 0 * R(2) \\
 &= R(2)
 \end{aligned}$$

10

であるから、 $D(1, 1) * D(2, 1)$ を計算すれば、 $R(2)$ が求まる。具体的に計算すると、 $D(1, 1)$ は00110110、 $D(2, 1)$ は00000011なので、 $R(2)$ は00110101となる。

【0087】

また同様に、

$$\begin{aligned}
 D(1, 2) * D(2, 2) &= (S(2) * R(1) * R(2)) * (S(2) * R(2)) \\
 &= (S(2) * S(2)) * R(1) * (R(2) * R(2)) \\
 &= 0 * R(1) * 0 \\
 &= R(1)
 \end{aligned}$$

であるから、 $D(1, 2) * D(2, 2)$ を計算すれば、 $R(1)$ が求まる。具体的に計算すると、 $D(1, 2)$ は10110011、 $D(2, 2)$ は00000010なので、 $R(1)$ は10110001となる。

20

【0088】

この $R(1), R(2)$ を使用して $S(1), S(2)$ を求める。

【0089】

$$\begin{aligned}
 D(2, 1) * R(1) &= (S(1) * R(1)) * R(1) \\
 &= S(1) * (R(1) * R(1)) \\
 &= S(1) * 0 \\
 &= S(1)
 \end{aligned}$$

であるから、 $D(2, 1) * R(1)$ を計算すれば、 $S(1)$ が求まる。具体的に計算すると、 $D(2, 1)$ は00000011、 $R(1)$ は10110001なので、 $S(1)$ は10110010となる。

【0090】

30

また同様に、

$$\begin{aligned}
 D(2, 2) * R(2) &= (S(2) * R(2)) * R(2) \\
 &= S(2) * (R(2) * R(2)) \\
 &= S(2) * 0 \\
 &= S(2)
 \end{aligned}$$

であるから $D(2, 2) * R(2)$ を計算すれば $S(2)$ が求まる。具体的に計算すると $D(2, 2)$ は00000001、 $R(2)$ は00110101なので、 $S(2)$ は00110111となる。

【0091】

一般に、 j を奇数として、

$$\begin{aligned}
 D(1, j) * D(2, j) &= (S(j) * R(j) * R(j+1)) * (S(j) * R(j)) \\
 &= (S(j) * S(j)) * (R(j) * R(j)) * R(j+1) \\
 &= 0 * 0 * R(j+1) \\
 &= R(j+1)
 \end{aligned}$$

40

であるから $D(1, j) * D(2, j)$ を計算すれば $R(j+1)$ が求まる。

【0092】

また同様に、

$$\begin{aligned}
 D(1, j+1) * D(2, j+1) &= (S(j+1) * R(j) * R(j+1)) * (S(j+1) * R(j+1)) \\
 &= (S(j+1) * S(j+1)) * R(j) * (R(j+1) * R(j+1)) \\
 &= 0 * R(j) * 0 \\
 &= R(j)
 \end{aligned}$$

50

であるから $D(1, j+1) \cdot D(2, j+1)$ を計算すれば $R(j)$ が求まる。

【0093】

この $R(j)$, $R(j+1)$ を使用して $S(j)$, $S(j+1)$ を求める。

【0094】

$$\begin{aligned} D(2, j) \cdot R(j) &= (S(j) \cdot R(j)) \cdot R(j) \\ &= S(j) \cdot (R(j) \cdot R(j)) \\ &= S(j) \cdot 0 \\ &= S(j) \end{aligned}$$

であるから $D(2, j) \cdot R(j)$ を計算すれば $S(j)$ が求まる。

【0095】

また同様に、

$$\begin{aligned} D(2, j+1) \cdot R(j+1) &= (S(j+1) \cdot R(j+1)) \cdot R(j+1) \\ &= S(j+1) \cdot (R(j+1) \cdot R(j+1)) \\ &= S(j+1) \cdot 0 \\ &= S(j+1) \end{aligned}$$

であるから $D(2, j+1) \cdot R(j+1)$ を計算すれば $S(j+1)$ が求まる。

【0096】

上述したように、元データの先頭から処理単位ビット長 b に基づいて分割処理を繰り返して行って、分割データを生成した場合には、3つの分割データ $D(1)$, $D(2)$, $D(3)$ のすべてを用いなくても、3つの分割データのうち、2つの分割データを用いて上述したように元データを復元することができる。

【0097】

本発明の他の実施形態として、乱数 R のビット長を元データ S のビット長よりも短いものを使用して、元データの分割処理を行うことができる。

【0098】

すなわち、上述した乱数 R は S , $D(1)$, $D(2)$, $D(3)$ と同じビット長のデータとしたが、乱数 R を元データ S のビット長より短いものとし、分割データ $D(1)$, $D(2)$, $D(3)$ の生成にこの短いビット長の乱数 R を繰り返し用いるものである。

【0099】

なお、分割データ $D(3)$ は乱数 R のみから生成されるので、分割データ $D(3)$ は乱数 R を繰り返して保管しておく必要はない。例えば、元データ S のビット長を 1600 ビット (200 バイト) としたとき、乱数 R は任意にとった 160 ビット (20 バイト) のデータの繰り返しとする。つまり、 $R(1) \sim R(20)$ はランダムに生成し、 $R(21) \sim R(200)$ は $R(21)=R(1)$, $R(22)=R(2)$, ..., $R(40)=R(20)$, $R(41)=R(1)$, $R(42)=R(2)$, ..., $R(60)=R(20)$, $R(61)=R(1)$, $R(62)=R(2)$, ..., $R(80)=R(20)$, ..., $R(181)=R(1)$, $R(182)=R(2)$, ..., $R(200)=R(20)$ とする。

【0100】

先の説明では、分割部分データ $D(3, j)$ を乱数部分データ $R(j)$ と定義して $D(3)$ を生成しているが、 $D(3, 20)$ まで保管すれば十分である。つまり、 $D(3)$ の長さは $D(1)$, $D(2)$ の 10 分の 1 となる。従って、保管すべきデータの総量は先の実施形態では元データ S の 3 倍であるが、この実施形態では 2.1 倍とすることができる。乱数 R における繰り返し部分のデータの長さは、短すぎると、 $D(1)$ または $D(2)$ のみから R が解読されてしまうことも考えられるため、適切な長さを選択することが望ましい。

【0101】

この実施形態では例えば乱数 R を生成するために疑似乱数生成アルゴリズムを使用する。乱数には自然界の物理現象などを使って乱数を発生させる真性乱数と、コンピュータのアルゴリズムなどで乱数を発生させる疑似乱数がある、真性乱数は、サイコロを何回も振ったり、雑音などの物理現象をりようしたりして生成することができるが、手間や装置がたいへんであるため、その代わりに、適当な長さの種 (乱数生成の種となる情報 (Seed)) から決定的なアルゴリズムに基づいて生成される疑似乱数が用いられる。例えば短い乱数を種とすれば長い乱数を得ることができる。種の長さは、例えば 128 ビット、

10

20

30

40

50

160ビットまたはそれ以上のものがある。決定的なアルゴリズムに基づいて生成されるといっても、統計的一様性、無相関性など乱数としてひつような性質を一定のレベルで満たしている。具体例としては、ANSI X9.42、FIPS 186-2など標準化されたものがある

(http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/cryptrec20030425_spec01.html)

【0102】

これらをもちいれば、乱数生成の種を入力として長い疑似乱数の列を生成することができる。例えば、160ビットの種を与えて元データSのビット長と同じ長さの乱数Rを生成し、上述したようにしてSとRからD(1)、D(2)を生成し、D(3)にはRを格納するのではなく160ビットの種を格納して保管することにより、元データSのビット長が大きくなってもD(3)に格納して保管すべきビット数は160ビットで済み、保管すべきデータの総量を押さえることができる。元データSを復元する場合には、D(3)に格納された160ビットの種から元データSのビット長と同じ長さの乱数Rを再度生成し、上述したようにして、これとD(1)またはD(2)を用いて元データSを復元することができる。

【0103】

上述した各実施形態は、元データを3つに分割し、そのうち2つから元データが復元可能となるものであったが、分割数nを3より大きくとって、n個より少ない個数の分割データから元データを復元することができることは勿論のことである。

【0104】

その1つの応用例として、元データを4つの分割データに分割する分割数n=4の場合の分割処理について図5に示すフローチャートおよび図6に示す定義式の一般形などを参照して説明する。

【0105】

まず、利用者は端末5から分割装置1にアクセスして元データSを送信し、分割装置1ではデータ送受信手段17が端末5からの元データSを受信し、分割装置1に供給する（ステップS301）。それから、利用者は端末5から分割数nとして4を分割装置1に指示する（ステップS303）。この分割数nは分割装置1において予め定められた値を用いてもよい。また、処理単位ビット長bが一例として8ビットと決定される（ステップS305）。次に、元データSのビット長が8×3の整数倍であるか否かを判定し、整数倍でない場合には、元データSの末尾を0で埋める（ステップS307）。また、整数倍を意味する変数mを0に設定する（ステップS309）。

【0106】

次に、元データSの8×3×m+1ビット目から8×3ビット分のデータが存在するか否かが判定される（ステップS311）。なお、本例では、元データSが8×3=24ビット長のデータの場合について説明している。

【0107】

ステップS311の判定の結果、本例の元データSでは8×3=24ビットのデータであり、変数m=1の場合に相当する8×3×m(=1)+1ビットに相当する25ビット以降のデータは存在しないので、ステップS311からステップS321に進むことになるが、今の場合は、変数mは0であるので、元データSの8×3×m+1ビット目は、8×3×0+1=1となり、元データSの24ビットの1ビット目から8×3ビット分にデータが存在するため、ステップS318に進む。

【0108】

ステップS318では、変数jを1から3(=分割数n-1)まで変えて、元データSの8×(3×m+j)+1ビット目から8ビット分(=処理単位ビット長)のデータを元部分データS(3×m+j)に設定し、これにより元データSを処理単位ビット長で分けした3個の元部分データS(1)、S(2)、S(3)が生成される。

【0109】

次に、変数jを1から3まで変えて、乱数部分データR(3×m+j)に乱数発生手段15から

10

20

30

40

50

発生する 8 ビットの長さの乱数を設定し、これにより乱数 R を処理単位ビット長で区分けした 3 個の乱数部分データ R(1), R(2), R(3) が生成される (ステップ S 315)。

【0110】

次に、ステップ S 317 において、変数 i を 1 から 4 (=分割数 n) まで変えるとともに、更に各変数 i において変数 j を 1 から 3 (=分割数 n-1) まで変えながら、ステップ S 317 に示す分割データを生成するための定義式により複数の分割データ D(i) の各々を構成する各分割部分データ D(i, 3×m+j) を生成する。この結果、次に示すような分割データ D が生成される。

【0111】

分割データ D

=4 個の分割データ D(i)=D(1), D(2), D(3), D(4)

第 1 の分割データ D(1)

=3 個の分割部分データ D(1, j)=D(1, 1), D(1, 2), D(1, 3)

第 2 の分割データ D(2)

=3 個の分割部分データ D(2, j)=D(2, 1), D(2, 2), D(2, 3)

第 3 の分割データ D(3)

=3 個の分割部分データ D(3, j)=D(3, 1), D(3, 2), D(3, 3)

第 4 の分割データ D(4)

=3 個の分割部分データ D(4, j)=D(4, 1), D(4, 2), D(4, 3)

なお、各分割部分データ D(i, j) を生成するためのステップ S 317 に示す定義式は、本例のように分割数 n=4 の場合には、具体的には図 6 に示す表に記載されているものとなる。図 6 に示す表から、分割部分データ D(1, 1) を生成するための定義式は $S(1)*R(1)*R(2)*R(3)$ であり、D(1, 2) の定義式は $S(2)*R(1)*R(2)*R(3)$ であり、D(1, 3) の定義式は $S(3)*R(1)*R(2)*R(3)$ であり、D(2, 1) の定義式は $S(1)*R(1)*R(2)$ であり、D(2, 2) の定義式は $S(2)*R(2)*R(3)$ であり、D(2, 3) の定義式は $S(3)*R(1)*R(3)$ であり、D(3, 1) の定義式は $S(1)*R(1)$ であり、D(3, 2) の定義式は $S(2)*R(2)$ であり、D(3, 3) の定義式は $S(3)*R(3)$ であり、D(4, 1) の定義式は $R(1)$ であり、D(4, 2) の定義式は $R(2)$ であり、D(4, 3) の定義式は $R(3)$ である。また、図 6 に示す表には $m>0$ の場合の任意の整数についての一般的な定義式も記載されている。

【0112】

このように変数 m=0 の場合について分割データ D を生成した後、次に変数 m を 1 増やし (ステップ S 319)、ステップ S 311 に戻り、変数 m=1 に該当する元データ S の 25 ビット以降について同様の分割処理を行おうとするが、本例の元データ S は 24 ビットであり、25 ビット以降のデータは存在しないので、ステップ S 311 からステップ S 321 に進み、上述したように生成した分割データ D(1), D(2), D(3), D(4) を分割装置 1 のデータ送受信手段 17 からネットワーク 3 を介して保管サーバ 7 にそれぞれ送信し、各保管サーバ 7 に保管し、分割処理を終了する。図 1 では保管サーバは 3 個であるが、分割数に応じて保管サーバを増やし、各分割データを異なる保管サーバに保管することが望ましい。

【0113】

ここで、上述した図 5 のフローチャートのステップ S 317 における定義式による分割データの生成処理、具体的には分割数 n=4 の場合の分割データの生成処理について詳しく説明する。

【0114】

まず、ステップ S 317 に示す定義式から各分割データ D(i)=D(1)~D(4) の各々を構成する各分割部分データ D(i, 3×m+j) は、次のようになる。

【0115】

$D(1, 3\times m+1)=S(3\times m+1)*Q(1, 1, 1)*Q(1, 1, 2)*Q(1, 1, 3)$

$D(1, 3\times m+2)=S(3\times m+2)*Q(2, 1, 1)*Q(2, 1, 2)*Q(2, 1, 3)$

$D(1, 3\times m+3)=S(3\times m+3)*Q(3, 1, 1)*Q(3, 1, 2)*Q(3, 1, 3)$

$D(2, 3\times m+1)=S(3\times m+1)*Q(1, 2, 1)*Q(1, 2, 2)*Q(1, 2, 3)$

$D(2, 3\times m+2)=S(3\times m+2)*Q(2, 2, 1)*Q(2, 2, 2)*Q(2, 2, 3)$

10

20

30

40

50

$$\begin{aligned}
D(2, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 2, 1) * Q(3, 2, 2) * Q(3, 2, 3) \\
D(3, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 3, 1) * Q(1, 3, 2) * Q(1, 3, 3) \\
D(3, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 3, 1) * Q(2, 3, 2) * Q(2, 3, 3) \\
D(3, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 3, 1) * Q(3, 3, 2) * Q(3, 3, 3) \\
D(4, 3 \times m + 1) &= R(3 \times m + 1) \\
D(4, 3 \times m + 2) &= R(3 \times m + 2) \\
D(4, 3 \times m + 3) &= R(3 \times m + 3)
\end{aligned}$$

次に、 $Q(j, i, k)$ を具体的に求める。これは $c(j, i, k)$ を 3×3 行列である $U[3, 3] \times (P[3, 3])^{(j-1)}$ の i 行 k 列の値としたとき下記のように定義される。

【0 1 1 6】

10

$c(j, i, k) = 1$ のとき $Q(j, i, k) = R(3 \times m + k)$

$c(j, i, k) = 0$ のとき $Q(j, i, k) = 0$

$j=1$ のときは

【数 1 1】

$$U[3, 3] \times (P[3, 3])^{(j-1)} = U[3, 3] \times (P[3, 3])^0 = U[3, 3]$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

20

【0 1 1 7】

$j=2$ のときは

【数 1 2】

$$U[3, 3] \times (P[3, 3])^{(j-1)} = U[3, 3] \times (P[3, 3])^1$$

$$= U[3, 3] \times (P[3, 3])$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

30

【0 1 1 8】

$j=3$ のときは

【数 1 3】

$$U[3, 3] \times (P[3, 3])^{(j-1)} = U[3, 3] \times (P[3, 3])^2$$

$$= U[3, 3] \times (P[3, 3]) \times (P[3, 3])$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

40

【0 1 1 9】

これを用いると、各分割部分データは次のような定義式により生成される。

50

【 0 1 2 0 】

$$\begin{aligned}
 D(1, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 1, 1) * Q(1, 1, 2) * Q(1, 1, 3) \\
 &= S(3 \times m + 1) * R(3 \times m + 1) * R(3 \times m + 2) * R(3 \times m + 3) \\
 D(1, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 1, 1) * Q(2, 1, 2) * Q(2, 1, 3) \\
 &= S(3 \times m + 2) * R(3 \times m + 1) * R(3 \times m + 2) * R(3 \times m + 3) \\
 D(1, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 1, 1) * Q(3, 1, 2) * Q(3, 1, 3) \\
 &= S(3 \times m + 3) * R(3 \times m + 1) * R(3 \times m + 2) * R(3 \times m + 3) \\
 D(2, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 2, 1) * Q(1, 2, 2) * Q(1, 2, 3) \\
 &= S(3 \times m + 1) * R(3 \times m + 1) * R(3 \times m + 2) \\
 D(2, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 2, 1) * Q(2, 2, 2) * Q(2, 2, 3) \\
 &= S(3 \times m + 2) * R(3 \times m + 2) * R(3 \times m + 3) \\
 D(2, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 2, 1) * Q(3, 2, 2) * Q(3, 2, 3) \\
 &= S(3 \times m + 3) * R(3 \times m + 1) * R(3 \times m + 3) \\
 D(3, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 3, 1) * Q(1, 3, 2) * Q(1, 3, 3) \\
 &= S(3 \times m + 1) * R(3 \times m + 1) \\
 D(3, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 3, 1) * Q(2, 3, 2) * Q(2, 3, 3) \\
 &= S(3 \times m + 2) * R(3 \times m + 2) \\
 D(3, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 3, 1) * Q(3, 3, 2) * Q(3, 3, 3) \\
 &= S(3 \times m + 3) * R(3 \times m + 3) \\
 D(4, 3 \times m + 1) &= R(3 \times m + 1) \\
 D(4, 3 \times m + 2) &= R(3 \times m + 2) \\
 D(4, 3 \times m + 3) &= R(3 \times m + 3)
 \end{aligned}$$

10

20

ここで、上述したように図 2 のステップ S 2 1 7 や図 5 のステップ S 3 1 7 で示した定義式に基づいて元データを分割する分割規則について一般的な表現で記載する。

【 0 1 2 1 】

まず、元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b で表すとともに、複数 n 個のうちの 1 つを表わす変数として $i (=1 \sim n)$ および $j (=1 \sim n-1)$ を用いて複数 $(n-1)$ 個の元部分データ、複数 $(n-1)$ 個の乱数部分データ、複数 (n) 個の分割データおよび各分割データの複数 $(n-1)$ 個の分割部分データのそれぞれのうちの 1 つをそれぞれ $S(j), R(j), D(i)$ および $D(i, j)$ で表わす。

30

【 0 1 2 2 】

それから、上記変数 j を 1 から $n-1$ まで変えて、各元部分データ $S(j)$ を元データ S の $b \times (j-1) + 1$ ビット目から b ビット分のデータとして作成する。次に、 $U[n, n]$ を $n \times n$ 行列である上三角行列とし、 $P[n, n]$ を $n \times n$ 行列である回転行列としたとき、 $c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times P[n-1, n-1]^T (j-1)$ の i 行 k 列の値と定義する。そして、 $c(j, i, k) = 1$ のとき、 $Q(j, i, k) = R(k)$ 、 $c(j, i, k) = 0$ のとき、 $Q(j, i, k) = 0$ と定義したとき、変数 i を 1 から n まで変えながら、各変数 i において変数 j を 1 から $n-1$ まで変えた場合において、 $i < n$ のとき、各分割部分データ $D(i, j)$ を

【 数 1 4 】

$$D(i, j) = S(j) * \left(\prod_{k=1}^{n-i} Q(j, i, k) \right)$$

40

【 0 1 2 3 】

と設定し、また $i = n$ のとき、各分割部分データ $D(i, j)$ を

$$D(i, j) = R(j)$$

と設定する。上記処理を元データ S の先頭から末尾まで繰り返し行うことにより元データ S から分割数 n の分割データを生成することができる。

【 0 1 2 4 】

次に、上述したように元データ S を 4 分割して生成された分割データ $D(1), D(2), D(3), D(4)$ から元データ S を復元する処理について図 6 を参照して説明する。なお、図 6 に示す 4

50

分割の場合には、変数 j を $3 \times m + 1$ ($m \geq 0$ である任意の整数) として、同図に示す一般的な定義式から次に示すように元データ S を生成することができる。

【0125】

まず、分割データ $D(1), D(2)$ から元データ S を求める場合について説明する。

【0126】

$$\begin{aligned} D(1, j) * D(2, j) &= (S(j) * R(j) * R(j+1) * R(j+2)) * ((S(j) * R(j) * R(j+1))) \\ &= (S(j) * S(j)) * (R(j) * R(j)) * (R(j+1) * R(j+1)) * R(j+2) \\ &= 0 * 0 * 0 * R(j+2) \\ &= R(j+2) \end{aligned}$$

従って、 $D(1, j) * D(2, j)$ を計算すれば、乱数 $R(j+2)$ が求まり、同様に $D(1, j+1) * D(2, j+1)$ を計算すれば、乱数 $R(j)$ が求まり、同様に $D(1, j+2) * D(2, j+2)$ を計算すれば、乱数 $R(j+1)$ が求まり、これらの得られた乱数 $R(j), R(j+1), R(j+2)$ を用いれば、

$$\begin{aligned} D(1, j) * R(j) * R(j+1) * R(j+2) &= (S(j) * R(j) * R(j+1) * R(j+2)) * (R(j) * R(j+1) * R(j+2)) \\ &= S(j) * (R(j) * R(j)) * (R(j+1) * R(j+1)) * (R(j+2) * R(j+2)) \\ &= S(j) * 0 * 0 * 0 \\ &= S(j) \end{aligned}$$

であるから、 $D(1, j) * R(j) * R(j+1) * R(j+2)$ を計算して、 $S(j)$ を求めてもよいし、 $D(2, j) * R(j) * R(j+1)$ から $S(j)$ を求めることもできる。

【0127】

同様に、 $D(1, j+1) * R(j) * R(j+1) * R(j+2)$ または $D(2, j+1) * R(j+1) * R(j+2)$ を計算して $S(j+1)$ を求めることができ、また同様に $D(1, j+2) * R(j) * R(j+1) * R(j+2)$ または $D(2, j+2) * R(j) * R(j+2)$ を計算して $S(j+2)$ を求めることができる。

【0128】

更に、上述したと同様に、 $D(2)$ と $D(3)$ から S を求めることができる。

【0129】

具体的には、まず $R(j), R(j+1), R(j+2)$ を求めてから、 $D(2, j), D(2, j+1), D(2, j+2)$ または $D(3, j), D(3, j+1), D(3, j+2)$ と $R(j), R(j+1), R(j+2)$ の XOR 演算により $S(j), S(j+1), S(j+2)$ を求めることができる。

【0130】

また更に、 $D(1)$ と $D(4)$ または $D(2)$ と $D(4)$ または $D(3)$ と $D(4)$ から S を求めることができる。

【0131】

$D(4)$ は R をそのものから定義したものであるから、計算することなく $D(4)$ から $R(j), R(j+1), R(j+2)$ を取得することができる、例えば、 $D(1, j), D(1, j+1), D(1, j+2)$ と $R(j), R(j+1), R(j+2)$ の XOR 演算により $S(j), S(j+1), S(j+2)$ を求めることができる。

【0132】

上述したように、演算回数の差が 1 である任意の 2 つの分割データ $D(1)$ と $D(2)$ 、または、 $D(2)$ と $D(3)$ 、または、 $D(4)$ と任意の 1 つの分割データ $D(1)$ または $D(2)$ または $D(3)$ から S が復元可能である。すなわち、4 つの分割データの中から任意に 3 つの分割データを取得すれば、その中には必ず上述したいずれかのケースが含まれるため、4 つのうち任意の 3 つの分割データから元データを復元可能である。

【0133】

図 7 は、5 分割の場合の分割データと定義式を示す表である。この 5 分割の場合は、 j を $4 \times m + 1$ (m は $m \geq 0$ である任意の整数) として、分割データの定義式から、上述した 4 分割の場合の復元処理と同様のことが言える。従って、演算回数の差が 1 である任意の 2 つの分割データ $D(1)$ と $D(2)$ 、または、 $D(2)$ と $D(3)$ 、または、 $D(3)$ と $D(4)$ 、または、 $D(5)$ と任意の 1 つの分割データ $D(1)$ または $D(2)$ または $D(3)$ または $D(4)$ から元データ S が復元可能である。そして、5 つの分割データの中から任意に 3 つの分割データを取得すれば、その中には必ずこのいずれかのケースが含まれるため、5 つのうち任意の 3 つから復元可能である

10

20

30

40

50

といえる。

【0134】

また、分割数 n を5より大きくとった場合も同様にして分割データを構成すれば、 n が奇数である場合は $(n+1)/2$ 個、 n が偶数である場合は $(n/2)+1$ 個の分割データから元データを復元することができる。この個数は、 n 個の分割データがあったときに、隣り合ったものを選択せず、かつ、 n 個目の分割データを選択しないような最大個数に1を加えたものである。つまり、前記最大個数に1を加えれば演算回数の差が1である2つの分割データまたは n 個目の分割データとその他のデータを必ず含むこととなるため、復元に必要な個数が前記のとおりといえる。

【0135】

次に、図8に示すフローチャートを参照して、分割数が n で、処理単位ビット長が b である場合の一般的な分割処理について説明する。

【0136】

まず、利用者は端末5から分割装置1にアクセスして元データ S を送信し、分割装置1ではデータ送受信手段17が端末5からの元データ S を受信し、分割装置1に供給する（ステップS401）。また、利用者は端末5から分割数 n （ $n \geq 3$ である任意の整数）を分割装置1に指示する（ステップS403）。この分割数 n は分割装置1において予め定められた値を用いてもよい。処理単位ビット長 b を決定する（ステップS405）。なお、 b は0より大きい任意の整数である。次に、元データ S のビット長が $b \times (n-1)$ の整数倍であるか否かを判定し、整数倍でない場合には、元データ S の末尾を0で埋める（ステップS407）。また、整数倍を意味する変数 m を0に設定する（ステップS409）。

【0137】

次に、元データ S の $b \times (n-1) \times m+1$ ビット目から $b \times (n-1)$ ビット分のデータが存在するか否かが判定される（ステップS411）。この判定の結果、データが存在しない場合は、ステップS421に進むことになるが、今の場合は、ステップS409で変数 m は0に設定された場合であるので、データが存在するため、ステップS413に進む。

【0138】

ステップS413では、変数 j を1から $n-1$ まで変えて、元データ S の $b \times ((n-1) \times m+j-1)+1$ ビット目から b ビット分のデータを元部分データ $S((n-1) \times m+j)$ に設定する処理を繰り返し、これにより元データ S を処理単位ビット長 b で区分けした $(n-1)$ 個の元部分データ $S(1), S(2), \dots, S(n-1)$ が生成される。

【0139】

次に、変数 j を1から $n-1$ まで変えて、乱数部分データ $R((n-1) \times m+j)$ に乱数発生手段15から発生する処理単位ビット長 b の乱数を設定し、これにより乱数 R を処理単位ビット長 b で区分けした $n-1$ 個の乱数部分データ $R(1), R(2), \dots, R(n-1)$ が生成される（ステップS415）。

【0140】

次に、ステップS417において、変数 i を1から n まで変えたとともに、更に各変数 i において変数 j を1から $n-1$ まで変えながら、ステップS417に示す分割データを生成するための定義式により複数の分割データ $D(i)$ の各々を構成する各分割部分データ $D(i, (n-1) \times m+j)$ を生成する。この結果、次に示すような分割データ D が生成される。

【0141】

分割データ D

= n 個の分割データ $D(i)=D(1), D(2), \dots, D(n)$

第1の分割データ $D(1)$

= $n-1$ 個の分割部分データ $D(1, j)=D(1, 1), D(1, 2), \dots, D(1, n-1)$

第2の分割データ $D(2)$

= $n-1$ 個の分割部分データ $D(2, j)=D(2, 1), D(2, 2), \dots, D(2, n-1)$

10

20

30

40

第 n の分割データ $D(n)$

$=n-1$ 個の分割部分データ $D(n, j)=D(n, 1), D(n, 2), \dots, D(n, n-1)$

このように変数 $m=0$ の場合について分割データ D を生成した後、次に変数 m を1増やし（ステップS419）、ステップS411に戻り、変数 $m=1$ に該当する元データ S の $b \times (n-1)$ ビット以降について同様の分割処理を行う。最後にステップS411の判定の結果、元データ S にデータがなくなった場合、ステップS411からステップS421に進み、上述したように生成した分割データ D を分割装置1のデータ送受信手段17からネットワーク3を介して保管サーバ7にそれぞれ送信し、各保管サーバ7に保管し、分割処理を終了する。図1では保管サーバは3個であるが、分割数に応じて保管サーバを増やし、各分割データを異なる保管サーバに保管することが望ましい。

10

【0142】

次に、図9に示すフローチャートを参照して、分割数 n が2の場合の分割処理について説明する。すなわち、上述した各実施形態は図8のフローチャートのステップS403に示したように分割数 n が3以上($n \geq 3$)の場合についてのものであるため、図9を用いて分割数 n が2の場合について説明する。

【0143】

まず、利用者は端末5から分割装置1にアクセスして元データ S を分割装置1に供給する（ステップS501）。また、利用者は端末5から分割数 n として2を分割装置1に指示する（ステップS503）。この分割数 n は分割装置1において予め定められた値を用いてもよい。それから処理単位ビット長 b として8ビットを決定する（ステップS505）。次に、元データ S のビット長が8の整数倍であるか否かを判定し、整数倍でない場合には、元データ S の末尾を0で埋める（ステップS507）。また、整数倍を意味する変数 m を0に設定する（ステップS509）。

20

【0144】

次に、元データ S の $8 \times m+1$ ビット目から8ビット分のデータが存在するか否かが判定される（ステップS511）。この判定の結果、データが存在しない場合は、ステップS521に進むことになるが、今の場合は、変数 m は0に設定されているので、データが存在するため、ステップS513に進む。

【0145】

ステップS513では、元データ S の $8 \times m+1$ ビット目から8ビット分のデータを元部分データ $S(m+1)$ に設定し、これにより元部分データ $S(1)$ が生成される。

30

【0146】

次に、乱数部分データ $R(m+1)$ に乱数発生手段15から発生する8ビットの乱数を設定し、これにより乱数部分データ $R(1)$ が生成される（ステップS515）。

【0147】

次に、ステップS517において、同ステップに示す定義式により分割データ D の各々を構成する各分割データ $D(1, m+1), D(2, m+1)$ が生成される。

【0148】

このように変数 $m=0$ の場合について分割データ D を生成した後、次に変数 m を1増やし（ステップS519）、ステップS511に戻り、変数 $m=1$ に該当する元データ S の8ビット以降について同様の分割処理を行う。最後にステップS511の判定の結果、元データ S にデータがなくなった場合、ステップS511からステップS521に進み、上述したように生成した分割データ $D(1)$ から $D(2)$ を分割装置1のデータ送受信手段17からネットワーク3を介して保管サーバ7にそれぞれ送信し、各保管サーバ7に保管し、分割処理を終了する。図1では保管サーバは3個であるが、このうち2個の保管サーバに各分割データを保管すればよい。

40

【0149】

ここにおいて、上述した図9のフローチャートのステップS517における定義式による分割データの生成処理、具体的には分割数 $n=2$ の場合の分割データの生成処理について詳しく説明する。

50

【0150】

変数 $m=0$ の場合には、ステップS517に示す定義式から各分割データ $D(1,1)$ 、 $D(2,1)$ は、次のようになる。

【0151】

$$D(1,1)=S(1)*Q(1,1,1)$$

$$D(2,1)=R(1)$$

次に、 $Q(j,i,k)$ を具体的に求める。ここで、 $n=2$ を定義に当てはめると、 j,i,k はいずれも1しか値をとらない。

【0152】

$c(j,i,k)$ は 1×1 行列である $U[1,1] \times (P[1,1])^{(j-1)}$ の i 行 k 列の値としたとき下記のよう

【0153】

$$c(j,i,k)=1 \text{ のとき } Q(j,i,k)=R(k)$$

$$c(j,i,k)=0 \text{ のとき } Q(j,i,k)=0$$

$$\begin{aligned} U[1,1] \times (P[1,1])^{(j-1)} &= U[1,1] \times (P[1,1])^0 \\ &= (1) \times E[1,1] \\ &= (1) \times (1) \\ &= (1) \end{aligned}$$

従って、 $c(1,1,1)$ は1であるから、 $Q(1,1,1)$ は $R(1)$ と定義される。

【0154】

以上から定義式は

$$D(1,1)=S(1)*R(1)$$

$$D(2,1)=R(1)$$

となる。変数 m を使用した形式では、

$$D(1,m+1)=S(m+1)*R(m+1)$$

$$D(2,m+1)=R(m+1)$$

となる。

【0155】

なお、分割数 $n=2$ の場合には、2個の分割データのうち、どちらか一方を取得しただけでは、元データ S を復元することはできず、2個のすべての分割データを取得して元データ S を復元することになる。

【0156】

さて、上述した実施形態においては、この分割データのみから、それを構成する部分データ間の演算を行うことによって乱数成分が失われる場合がある。即ち、例えば3分割の場合、各分割部分データは次のように定義される。

【0157】

$$D(1,1)=S(1)*R(1)*R(2), \quad D(1,2)=S(2)*R(1)*R(2),$$

$$D(2,1)=S(1)*R(1), \quad D(2,2)=S(2)*R(2),$$

$$D(3,1)=R(1), \quad D(3,2)=R(2).$$

$D(1)$ について見ると、例えば、 $D(1,1)$ 、 $D(1,2)$ が取得できると、

$$\begin{aligned} D(1,1)*D(1,2) &= (S(1)*R(1)*R(2))*(S(2)*R(1)*R(2)) \\ &= S(1)*S(2)*((R(1)*R(1))*(R(2)*R(2))) \\ &= S(1)*S(2)*0*0 \\ &= S(1)*S(2) \end{aligned}$$

となる。一般には $D(1,j)*D(1,j+1)=S(j)*S(j+1)$ である。ここで j は $j=2 \times m + 1$ 、 m は $m \geq 0$ の任意の整数である。

【0158】

$D(1,1)$ 、 $D(1,2)$ は、上記の定義より、元データと乱数の演算により生成されたものであり、 $D(1,1)$ 、 $D(1,2)$ それぞれを見ても元データの内容は分からないが、 $D(1,1)*D(1,2)$ の演算を行うことにより $S(1)*S(2)$ が算出される。これは元データそのものではないが、

乱数成分を含んでいない。

【0159】

このように乱数成分が失われると、個々の元部分データについて、例えば $S(2)$ の一部が既知である場合には $S(1)$ の一部が復元可能となるので、安全ではないと考えられる。例えば、元データが標準化されたデータフォーマットに従ったデータであって、 $S(2)$ がそのデータフォーマット中のヘッダ情報やパディング（例えば、データ領域の一部を0で埋めたもの）などを含む部分であった場合には、これらのデータフォーマット固有のキーワードや固定文字列などを含むため、その内容は予測され得る。また、 $S(2)$ のうち既知の部分と $S(1)*S(2)$ の値から、 $S(1)$ の一部が復元可能である。

【0160】

4分割の場合は、図6より、 $D(2, j)*D(2, j+1)*D(2, j+2)=S(j)*S(j+1)*S(j+2)$ である。ここで j は $j = 3 \times m + 1$ 、 m は $m \geq 0$ の任意の整数である。

【0161】

5分割の場合は、図7より、 $D(i, j)*D(i, j+1)*D(i, j+2)*D(i, j+3)=S(j)*S(j+1)*S(j+2)*S(j+3)$ である。ここで i は1または3、 j は $j = 4 \times m + 1$ 、 m は $m \geq 0$ の任意の整数である。

【0162】

分割数が5より大きい場合も同様に演算により、乱数成分が失われる。なお、分割数が2の場合にはこのような問題は生じない。

【0163】

この問題を解決する一つの方法は以下の通りである。これは3分割の場合に適用可能な方法である。図10における $D(1, j+1)$ と $D(2, j+1)$ は、図4における $D(1, j+1)$ と $D(2, j+1)$ を入れ替えたものである。ここで j は $j = 2 \times m + 1$ 、 m は $m \geq 0$ の任意の整数である。

【0164】

この場合、個々の分割データのみでは、それを構成する分割部分データ間で演算を行っても乱数成分が失われず、これは、図10より

$$\begin{aligned} D(1, j)*D(1, j+1) &= (S(j)*R(j)*R(j+1))*(S(j+1)*R(j+1)) \\ &= S(j)*S(j+1)*R(j)*((R(j+1)*R(j+1))) \\ &= S(j)*S(j+1)*R(j)*0 \\ &= S(j)*S(j+1)*R(j) \\ D(2, j)*D(2, j+1) &= (S(j)*R(j))*(S(j+1)*R(j)*R(j+1)) \\ &= S(j)*S(j+1)*(R(j)*R(j))*R(j+1) \\ &= S(j)*S(j+1)*0*R(j+1) \\ &= S(j)*S(j+1)*R(j+1) \end{aligned}$$

$$D(3, j)*D(3, j+1) = R(j)*R(j+1)$$

となるからである。

【0165】

また、この場合、3つの分割データのうち2つから、元データを復元することができるという特性は失われていない。これは、 $D(1)$ 、 $D(2)$ を取得して S を復元する場合には、図10における $D(1)$ 、 $D(2)$ は、図4における $D(1)$ 、 $D(2)$ を構成する分割部分データを入れ替えたものにすぎないので、明らかにこれらから元データを復元することができ、また、 $D(1)$ と $D(3)$ または $D(2)$ と $D(3)$ を取得して S を復元する場合には、 $D(3)$ は乱数のみからなる分割データであるので、 $D(1)$ または $D(2)$ の分割部分データ毎に必要な個数の乱数との排他的論理和演算を行うことにより、乱数部分を消去して元データを復元することができるからである。

【0166】

上記の問題を解決する別の方法は以下の通りである。これは分割数が3以上である場合に分割数には関係なく適用可能な方法である。図11、図12、図13は、図4、図6、図7において $D(i, j)$ を生成する個々の定義式から $R(j)$ を削除したものである。ここで i は $n-1 > i > 0$ であり、 j は $j = (n-1) \times m + 1$ 、 m は $m \geq 0$ の任意の整数、 n は分割数である。

これは分割数が5より大きい場合でも同様である。

【0167】

この場合も、個々の分割データのみでは、それを構成する分割部分データ間で演算を行っても乱数成分が失われない。これは、図4、図6、図7においては、個々の分割データの分割部分データ間で演算をすると乱数部分が消去されて、3分割の場合には $D(1, j) * D(1, j+1) = S(j) * S(j+1)$ (j は $j = 2 \times m + 1$ 、 m は $m \geq 0$ の任意の整数) となり、4分割の場合には $D(2, j) * D(2, j+1) * D(2, j+2) = S(j) * (S(j+1) * S(j+2))$ (j は $j = 3 \times m + 1$ 、 m は $m \geq 0$ の任意の整数) となり、5分割の場合には $D(i, j) * D(i, j+1) * D(i, j+2) * D(i, j+3) = S(j) * S(j+1) * S(j+2) * S(j+3)$ (i は1または3、 j は $j = 4 \times m + 1$ 、 m は $m \geq 0$ の任意の整数) となっていたが、上記の通り $D(i, j)$ を生成する個々の定義式から $R(j)$ を削除した (i は $n-1 > i > 0$ 、 j は $j = (n-1) \times m + 1$ 、 m は $m \geq 0$ の任意の整数、 n は分割数) ので、一つの $R(j)$ が消去されずに残ることになるためである。

10

【0168】

また、この場合も、 n 個の分割データのうちの所定の個数の分割データから、元データを復元することができるといった特性は失われていない。

【0169】

まず3分割の場合には、 $D(1)$ 、 $D(2)$ を取得して S を復元する場合は、上述した通り $R(j)$ (j は $j = 2 \times m + 1$ 、 m は $m \geq 0$ の任意の整数) が $D(1, j) * D(2, j+1)$ から求まり、 $S(j)$ が $D(2, j) * R(j) = S(j) * R(j) * R(j)$

$$= S(j) * 0$$

$$= S(j)$$

20

から求まり、 $R(j+1)$ が

$$D(1, j) * S(j) = S(j) * R(j+1) * S(j)$$

$$= S(j) * S(j) * R(j+1)$$

$$= 0 * R(j+1)$$

$$= R(j+1)$$

から求まり、上述した通り $S(j+1)$ が $D(2, j+1) * R(j+1)$ から求まる。また、 $D(1)$ と $D(3)$ または $D(2)$ と $D(3)$ を取得して S を復元する場合には、 $D(3)$ は乱数のみからなる分割データであるので、 $D(1)$ または $D(2)$ の分割部分データ毎に必要な個数の乱数との排他的論理和演算を行うことにより、乱数部分を消去して元データを復元することができる。

30

【0170】

次に4分割の場合には、 $D(1)$ 、 $D(2)$ を取得して S を復元する場合は、 $R(j+2)$ が

$$D(1, j) * D(2, j) = (S(j) * R(j+1) * R(j+2)) * (S(j) * R(j+1))$$

$$= (S(j) * S(j)) * (R(j+1) * R(j+1)) * R(j+2)$$

$$= 0 * 0 * R(j+2)$$

$$= R(j+2)$$

から求まり、上述した通り $S(j)$ が $D(1, j) * R(j+1) * R(j+2)$ または $D(2, j) * R(j+1)$ から求まる。

【0171】

また、 $D(2)$ と $D(3)$ を取得して S を復元する場合には、上述した通り $R(j+2)$ が $D(2, j+1) * D(3, j+1)$ から求まり、 $R(j)$ が $D(2, j+2) * D(3, j+2)$ から求まり、 $S(j)$ が

40

$$D(3, j) * R(j) = (S(j) * R(j)) * R(j)$$

$$= S(j) * (R(j) * R(j))$$

$$= S(j) * 0$$

$$= S(j)$$

から求まり、 $R(j+1)$ が

$$D(2, j) * S(j) = (S(j) * R(j+1)) * S(j)$$

$$= (S(j) * S(j)) * R(j+1)$$

$$= 0 * R(j+1)$$

$$= R(j+1)$$

50

から求まり、上述した通り $S(j)$ が $D(1, j) \cdot R(j+1) \cdot R(j+2)$ または $D(2, j) \cdot R(j+1)$ から求まる。

【0172】

また、 $D(4)$ と任意の一つの分割データ $D(1)$ または $D(2)$ または $D(3)$ を取得して S を復元する場合には、 $D(4)$ は乱数のみからなる分割データであるので、 $D(1)$ または $D(2)$ または $D(3)$ の分割部分データ毎に必要な個数の乱数との排他的論理和演算を行うことにより、乱数部分を消去して元データを復元することができる。

【0173】

従って、演算回数の差が1である任意の2つの分割データ $D(1)$ と $D(2)$ 、または、 $D(2)$ と $D(3)$ 、または、 $D(4)$ と任意の1つの分割データ $D(1)$ または $D(2)$ または $D(3)$ から S が復元可能である。すなわち、4つの分割データの中から任意に3つの分割データを取得すれば、その中には必ず上述したいずれかのケースが含まれるため、4つのうち任意の3つの分割データから元データを復元可能である。

【0174】

次に5分割の場合については、 $D(1)$ と $D(2)$ または $D(2)$ と $D(3)$ を取得して S を復元する場合、 $D(3)$ と $D(4)$ を取得して S を復元する場合、 $D(5)$ と任意の1つの分割データ $D(1)$ または $D(2)$ または $D(3)$ または $D(4)$ を取得して S を復元する場合のいずれも4分割の場合と同様である。

【0175】

従って、演算回数の差が1である任意の2つの分割データ $D(1)$ と $D(2)$ 、または、 $D(2)$ と $D(3)$ 、または、 $D(3)$ と $D(4)$ 、または、 $D(5)$ と任意の1つの分割データ $D(1)$ または $D(2)$ または $D(3)$ または $D(4)$ から元データ S が復元可能である。そして、5つの分割データの中から任意に3つの分割データを取得すれば、その中には必ずこのいずれかのケースが含まれるため、5つのうち任意の3つから復元可能であるといえる。

【0176】

また、分割数 n を5より大きくとった場合も同様にして分割データを構成すれば、 n が奇数である場合は $(n+1)/2$ 個、 n が偶数である場合は $(n/2)+1$ 個の分割データから元データを復元することができる。この個数は、 n 個の分割データがあったときに、隣り合ったものを選択せず、かつ、 n 個目の分割データを選択しないような最大個数に1を加えたものである。つまり、前記最大個数に1を加えれば演算回数の差が1である2つの分割データまたは n 個目の分割データとその他のデータを必ず含むこととなるため、復元に必要な個数が前記のとおりといえる。

【0177】

なお、上記実施形態のデータ分割方法の処理手順をプログラムとして例えばCDやFDなどの記録媒体に記録して、この記録媒体をコンピュータシステムに組み込んだり、または記録媒体に記録されたプログラムを通信回線を介してコンピュータシステムにダウンロードしたり、または記録媒体からインストールし、該プログラムでコンピュータシステムを作動させることにより、データ分割方法を実施するデータ分割装置として機能させることができることは勿論であり、このような記録媒体を用いることにより、その流通性を高めることができるものである。

【0178】

上述してきたように、本実施例によれば、所定の定義式が元部分データと乱数部分データの排他的論理和からなるので、従来のように多項式や剰余演算を行う高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができる。

【0179】

また、生成した複数の分割データのうち分割数よりも少ない数の分割データに対して定義式を適用することにより元データを復元するので、分割数よりも少ない任意の数 x の分割データで元データを復元でき、分割数から x を減算した数までの分割データを紛失したり破壊したとしても、元データを復元することができる。

【0180】

さらに、元データをネットワークを介して端末から受信し、この元データに対して元部分データ、乱数部分データおよび分割部分データの生成処理を施して生成された複数の分割部分データをネットワークを介して保管サーバに送信して保管するので、多数のユーザが端末からネットワークを介してアクセスして分割処理を依頼することができ、共通化および経済化を図ることができる。

【0181】

なお、上述した実施形態は、分割データは、乱数のみからなる1つの分割データと、1つの元部分データと1つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる1つ以上の分割データを含む場合であるが、上述した実施形態を変形して分割データは、乱数のみからなる1つ以上の分割データと、1つ以上の元部分データと1つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる1つ以上の分割データを含むものとしても良い。また、上述した実施形態を変形して、分割データは、1つ以上の元部分データと1つ以上の乱数部分データの排他的論理和演算によって生成された分割部分データからなる2つ以上の分割データを含むものとしても良い。

【図面の簡単な説明】

【0182】

【図1】本発明の一実施形態に係るデータ分割方法を実施するデータ分割装置を含むシステム構成図である。

【図2】図1に示す実施形態のデータ分割装置の分割数 $n=3$ の場合の分割処理を示すフローチャートである。

【図3】16ビットの元データ S を8ビットの処理単位ビット長に基づいて分割数 $n=3$ で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【図4】分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図5】図1に示す実施形態のデータ分割装置の分割数 $n=4$ の場合の分割処理を示すフローチャートである。

【図6】分割数 $n=4$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図7】分割数 $n=5$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図8】図1に示す実施形態のデータ分割装置の分割数が n で処理単位ビット長が b である場合の一般的な分割処理を示すフローチャートである。

【図9】図1に示す実施形態のデータ分割装置の分割数が2である場合の分割処理を示すフローチャートである。

【図10】分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式の別の例を示す表である。

【図11】分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式の更に別の例を示す表である。

【図12】分割数 $n=4$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式の別の例を示す表である。

【図13】分割数 $n=5$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式の別の例を示す表である。

【符号の説明】

【0183】

- 1 分割装置
- 3 ネットワーク
- 5 端末

10

20

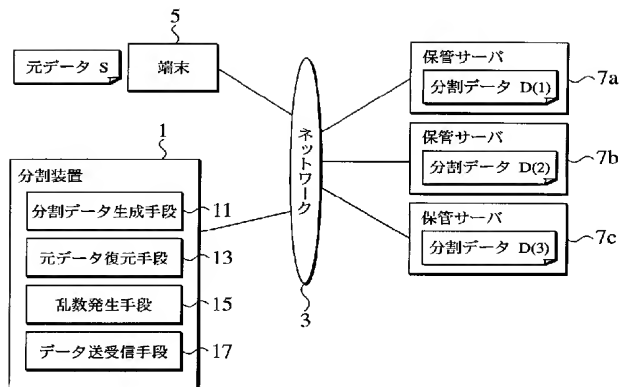
30

40

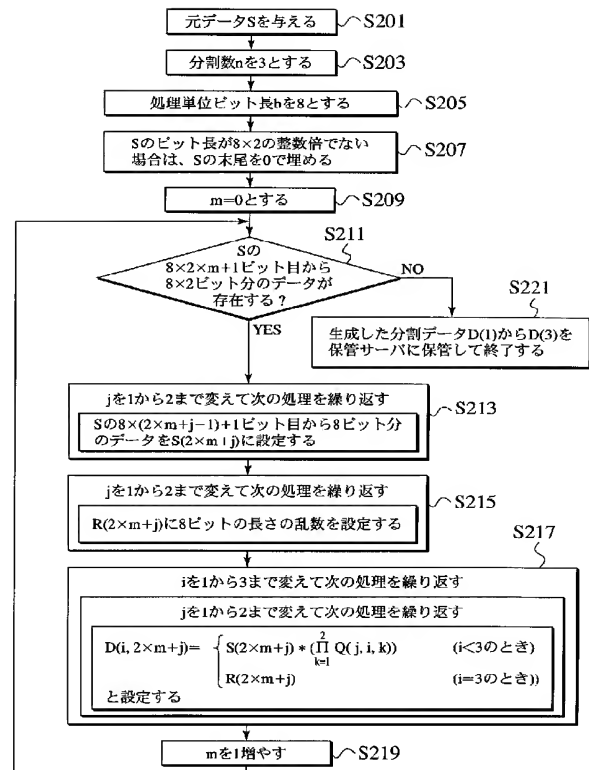
50

- 7a, 7b, 7c 保管サーバ
 11 分割データ生成手段
 13 元データ復元手段
 15 乱数発生手段
 17 データ送受信手段

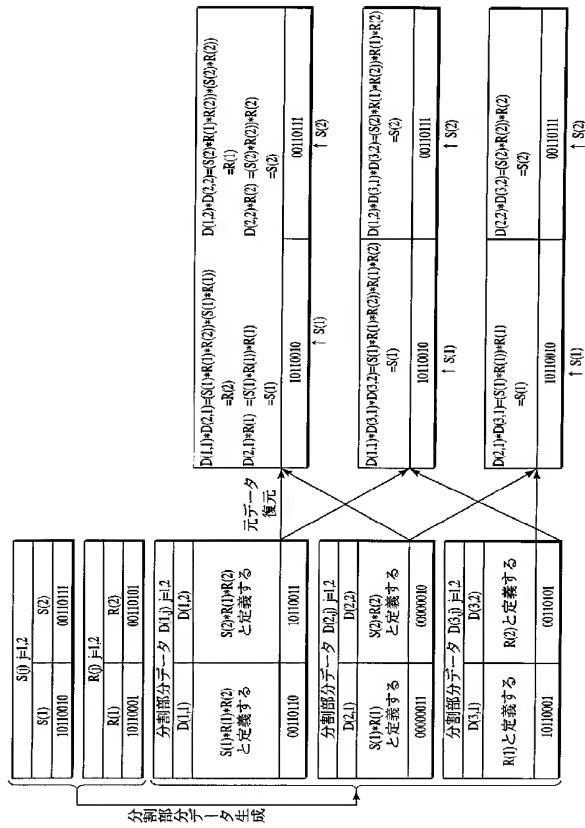
【図1】



【図2】



【図3】



【図4】

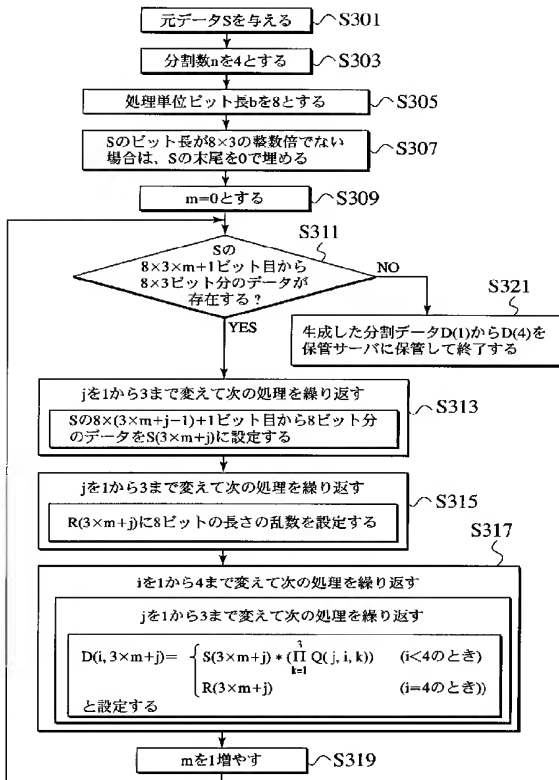
3分割 ($n=3$)
任意の2つの分割データから元データが復元可能

(m は $m>0$ の任意の整数)

j の値	1	2	...	$j=2 \times m+1$	$j+1$...
元データ $S(j)$	$S(1)$	$S(2)$...	$S(j)$	$S(j+1)$...
乱数 $R(j)$	$R(1)$	$R(2)$...	$R(j)$	$R(j+1)$...
分割部分データ $D(1,j)$	$S(1) \neq R(1)$	$S(2) \neq R(2)$...	$S(j) \neq R(j)$	$S(j+1) \neq R(j+1)$...
分割部分データ $D(2,j)$	$S(1) \neq R(1)$	$S(2) \neq R(2)$...	$S(j) \neq R(j)$	$S(j+1) \neq R(j+1)$...
分割部分データ $D(3,j)$	$R(1)$	$R(2)$...	$R(j)$	$R(j+1)$...

→元データSの末尾まで続く

【図5】



【図6】

4分割 ($n=4$)
任意の3つの分割データ(取り方によっては2つの分割データ)から元データが復元可能

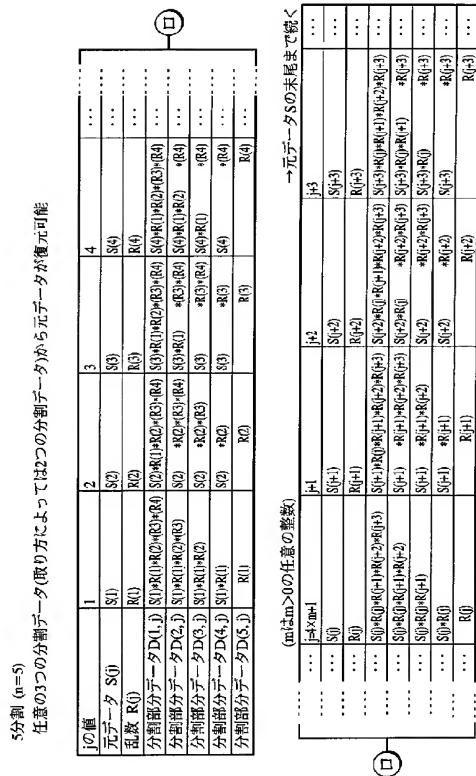
(m は $m>0$ の任意の整数)

j の値	1	2	3	...
元データ $S(j)$	$S(1)$	$S(2)$	$S(3)$...
乱数 $R(j)$	$R(1)$	$R(2)$	$R(3)$...
分割部分データ $D(1,j)$	$S(1) \neq R(1)$	$S(2) \neq R(2)$	$S(3) \neq R(3)$...
分割部分データ $D(2,j)$	$S(1) \neq R(1)$	$S(2) \neq R(2)$	$S(3) \neq R(3)$...
分割部分データ $D(3,j)$	$S(1) \neq R(1)$	$S(2) \neq R(2)$	$S(3) \neq R(3)$...
分割部分データ $D(4,j)$	$R(1)$	$R(2)$	$R(3)$...

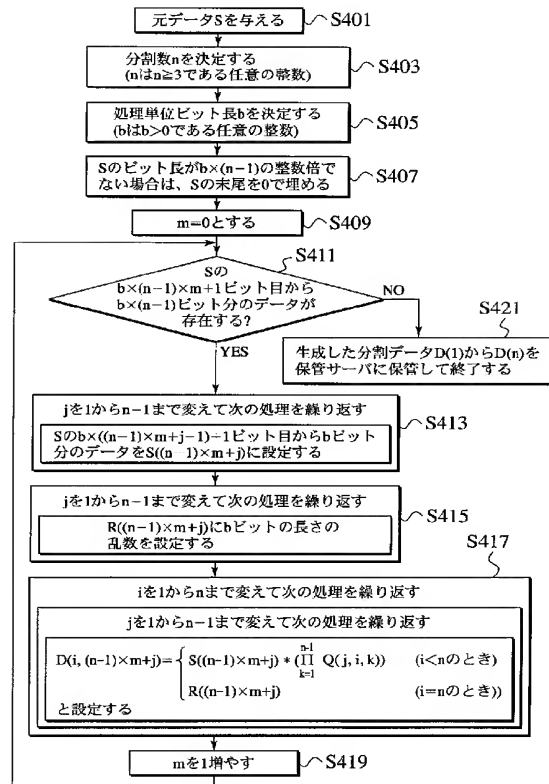
→元データSの末尾まで続く

...	$j=3 \times m+1$	$j+1$	$j+2$...
...	$S(j)$	$S(j+1)$	$S(j+2)$...
...	$R(j)$	$R(j+1)$	$R(j+2)$...
...	$S(j) \neq R(j)$	$S(j+1) \neq R(j+1)$	$S(j+2) \neq R(j+2)$...
...	$S(j) \neq R(j)$	$S(j+1) \neq R(j+1)$	$S(j+2) \neq R(j+2)$...
...	$S(j) \neq R(j)$	$S(j+1) \neq R(j+1)$	$S(j+2) \neq R(j+2)$...
...	$R(j)$	$R(j+1)$	$R(j+2)$...

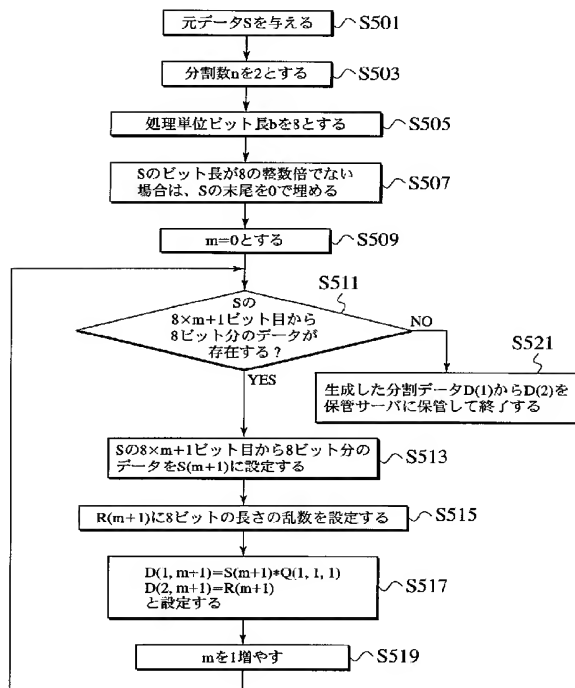
【図7】



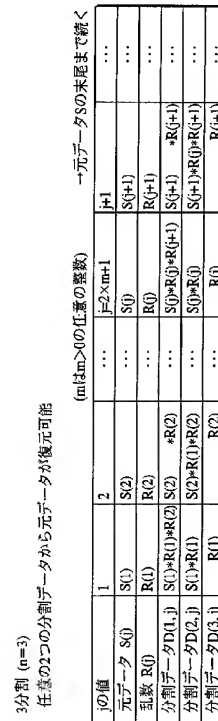
【図8】



【図9】



【図10】



フロントページの続き

(72)発明者 荻原 利彦

東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内

(72)発明者 野村 進

東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内

Fターム(参考) 5B017 AA03 BA10 CA15 CA16